DOCUMENT RESUME

ED 110 006 IR 002 274

AUTHOR	Thomas, Charles R., Ed.; Manning, Sherry, Ed.
TITLE	The Future of Higher Education Information Systems: Yesterday's Solutions are Tomorrow's Problems. Proceedings of the 1973 CAUSE National Conference, New Orleans, December, 1973.
INSTITUTION	College and Univ. Systems Exchange, Boulder, Colo.
PUB DATE	74
NOTE	829p.
EDRS PRICE	MF-$1.39 HC-$42.53 PLUS POSTAGE
DESCRIPTORS	*Administration; College Administration; Colleges; Computer Oriented Programs; *Computer Programs; Conference Reports; Data Bases; Data Collection; *Data Processing; Electronic Data Processing; *Higher Education; Information Systems; Management Information Systems; *Management Systems; Systems Analysis; Systems Concepts; Systems Development; Universities; University Administration
IDENTIFIERS	Administrative Data Processing; CAUSE; *College and University Systems Exchange

ABSTRACT

Fifty-nine addresses, technical papers, and descriptions of institutional administrative data processing (ADP) applications presented at the 1973 College and University Systems Exchange (CAUSE) National Conference are compiled. Topics include (1) operational data systems, (2) management information systems, (3) planning and management systems, (4) new developments in computer software for instructional and ADP operations in colleges and universities, and (5) specialized ADP applications for higher education. (DGC)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED110006

The Future of Higher Education Information Systems:

Yesterday's Solutions are Tomorrow's Problems.

Proceedings of the
1973 CAUSE National Conference

NEW ORLEANS

DECEMBER, 1973

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

edited by
Charles R. Thomas
and
Sherry Manning

IR 002 274

2/3

CAUSE College And University Systems Exchange
737 Twenty-Ninth Street, Boulder, Colorado 80303
(303) 447-9441

ACKNOWLEDGMENTS

The College and University Systems Exchange expresses appreciation to each one of the participants in the 1973 National Conference Program. The participants represented both the administration and faculty of colleges and universities as well as researchers, consultants, and manufacturers of computer hardware, software and support systems. The diversity of backgrounds, experiences, and perspectives of the participants contributed significantly to the success of the Conference and to the improved future of higher education information systems.

CAUSE especially acknowledges the financial support provided to the Conference by organizations. A special contribution by IBM has assisted with the publication of the *Proceedings* of the Conference. In addition, the coffee breaks and social hours were sponsored by supporting organizations. CAUSE thanks and acknowledges the companies and their representatives here:

Burroughs Corporation
Gordon W. Mills
Charles A. Pickett

Digital Equipment Corp.
Robert Good
Alice Peters

Informatics, Inc.
John Bentivegna
Tony Lamia

3M Corporation
Carol Bartz
Steven R. Longaker

National Cash Register Co.
Thomas Dodderidge

Program Products, Inc.
Sherman A. Drusin
Paul R. Kalbach

Systems & Computer Technology
Jack F. Magosin

Xerox Corporation
Carl Hoffman

The College and University Systems Exchange also expresses a special thanks to the members of the Program Committee for their efforts in organizing, planning and hosting the National Conference. They are:

Robert E. Taylor, Chairman
Virginia Polytechnic Institute & State Univ.

Louis R. Leurig
University of New Mexico

Michael E. Schouest
Louisiana State University

Henry G. Vaughan
Cornell University

Bartley J. Carlson
Waubonsee Community College

Paul J. Plourde
Amherst College

Donald Tiernan
Northern Iowa University

Lastly, CAUSE is indebted to Deborah Smith for the many hours she spent preparing this publication.

INTRODUCTION

Conferences and symposia serve many functions to the participants and to the state of the art of the profession. The opportunity to present and listen to new ideas and applications in a structured way is excellent; the chance for professionals in like positions to compare similar problems, as well as the opportunities of cross-fertilization of ideas among professionals in complementing fields, is of great benefit. Often understated is the intangible benefit one professional gains in leaving operational problems at home and having a chance to reflect on broader and more long-run issues in a fresh environment.

The College and University Systems Exchange hopes that the 1973 Conference provided such benefits to its participants and hopes that the state of the art of higher education information systems has been advanced.

With the theme, "The Future of Higher Education Information Systems: Yesterday's Solutions are Tomorrows's Problems," the College and University Systems Exchange launches a new era. It is hoped that this era will be marked with by an increased sharing of ideas, systems, and experiences; a new maturity and professionalism; and the development of a forum and continuing dialogue of ideas and problem solutions to our members and readership. The Proceedings of the CAUSE National Conference reports the efforts of many to establish the first such forum. Let the dialogue begin.

Robert E. Taylor
Conference Chairman

COLLEGE AND UNIVERSITY SYSTEMS EXCHANGE
NATIONAL CONFERENCE PROCEEDINGS

New Orleans, Louisiana December 10-12, 1973

TABLE OF CONTENTS

Page

The Million Multipliers
William J. McKeefery 1

Computer Languages of the Future
David B. Ward 9

To COM or Not to COM
Ernest L. Jones 17

CAUSE DBMS Project Report
Paul J. Plourde and Bard F. White 37

DBMS DELPHI Study
Bard F. White 45

Experience in the Development of Management Information Systems
Donald C. Bruegman 65

Virginia Tech Administrative Management Information System
A. Wayne Donald 83

Designing Large Networks
Craig Johnson 125

An Educational Computer Users Network
Thomas H. Beerman and Gregory S. Jonesku 127

A Data Base Is Not A System
Wallace C. Knapp and Robert G. Ogilvie 147

MARK IV in the Data Base Environment
Tony Lamia 149

Are We Teaching Computer Science or Computer Scientists?
Alan Taylor 151

C6

Page

Executive Perspectives on MIS
John L. Green, Thomas S. Smith, Henry G. Vaughan, Herbert Zeitlin 153

Planning and Control of Development
Herbert W. Bomzer 165

A Data Set Management System
Edward M. Staman and B. E. Squires 167

DBMS Evaluation Panel
Kenneth E. Shostack, John L. Sullivan, R. Brian Walsh and L. Carvel Whiting 189

Computer Performance Measurement and Evaluation A Perspective
Tad B. Pinkerton 241

Future Trends in System Architecture
Arthur E. Scott 247

Feature Analysis: Student Financial Systems
Dominic Bordonaro and R. Peter Jackson 249

Keynote Address: APL and Its Use in Education
Kenneth E. Iverson 267

Inter-Institutional Systems Development
Jack W. Smith 269

ADP Uses of Mini-Computers
Lloyd Michalsen and Alice Peters 293

Feature Analysis: Alumni/Development Systems
Roger N. Gaunt and David P. Tursan 299

Data Independent Computer Programs and Information Retrieval
T. Ray Nanney 325

Implementing a Cooperative Computing Plan
David F. Nyman 355

Connecticut's Information System Plan
Francis J. Degnan 357

A Statewide Approach to Management Information Systems for Vocational and Technical Institutions
K. M. Hussain and D. M. Nowrasteh 365

Page

Point of Service Data Recording: If It's Here, Where Is It Going?
Thomas E. Doddridge 385

Feature Analysis: CORE Financial System
Dennis M. Klinger 387

ADMINS: Data Management on a Mini-Computer
David Griffel and Stuart McIntosh 397

Administrative Applications of APL
Adin Falkoff 415

Administrative Uses of APL at Syracuse University
Phyllis Kent 419

Non-Numeric Use of APL in the University Environment
L. P. A. Robichaud 439

Using APL to Provide a Computer Utility for Education
Rebecca E. Willis 441

An On-Line MIS to Assist the Management of Facilities
S. C. Fallis, D. D. McElroy and E. E. Payne . . 447

An APL Solution to a Data Processing Problem
Robert E. Schaulis 457

Interactive Analysis of Student Records
P. C. Berry and Richard M. McCarthy 459

Security Facilities in an APL System
Richard Cox 461

Feature Analysis: Student Records Systems
Verner R. Ekstrom 471

Feature Analysis: Payroll/Personnel Systems
L. Carvel Whiting 515

Separatism Revisited
Michael M. Roberts 529

Combined Computing Panel
Robert C. Heterick, Gordon R. Sherman,
Martin Solomon and Donald J. Wright 531

Text Processing at the Virginia Tech Computing Center
Ellis P. Bucklen 533

Page

OS-OASIS Conversion
Roger C. Lauzon 557

Software Design and Architecture
James W. Boyd 577

Toward a National Education Data Base:
Experience of the National Commission on the
Financing of Postsecondary Education
Daryl E. Carlson, James Farmer and
Richard E. Stanton 613

Data Entry Techniques
Haymond H. Satterfield 661

National Survey on Computer Based MIS in Higher
Education
Richard L. Mann 663

Rounding Second Base with Data Base
Glenn B. Harvey 675

Data Base Dictionary Design
Paul J. Plourde 699

An ENVIRON/1 Administrative On-Line System
David Clements 717

Data Entry Practices Survey
Harry Grothjahn, Dennis Klinger, Frank Martin
and Wayne Ostendorf 719

Software Procurement Panel
Robert C. Heterick, Ralph E. Lee, Richard T.
Paumen and Henry G. Vaughan 721

The Use of APL at a New and Emerging Institution
C. William Higginbotham and Sampson G. Rice . . 723

The Design of a Generalized Mini-Computer Based
College Information System
George A. Mapp 739

CODASYL Data Base Report
James Gilliam 763

Data Base Administration: Theory Into Practice
Jane Ryland 781

Feature Analysis: Registration Schedulers
Vinod Chachra 791

System 2000 As a DBMS for Colleges and Universities
Charles J. Lewis 827

KEYNOTE ADDRESS

THE MILLION MULTIPLIERS

William J. McKeefery
President
The William Paterson College
New Jersey

The Million Multipliers

George Bernard Shaw once said, The only time a person's education is interrupted is when he goes to school. What I think he meant is that the unscheduled, unusual and unexpected happening often may provide a rich learning experience. Cicero related that conferences or symposia as he knew them were successful if all men would come together and lay their problems in one common heap; then most would be content to take back their own and depart. This meeting in New Orleans where French, Spanish and Early American traditions are intertwined, seems to be a good place to look at the computer business in a fresh way. As you may guess from the title, I hope to draw some ideas together for purpose of cross-fertilization. We often become so preoccupied with all that is new, that we miss the contrasts.

The other day two directors of computing centers took a few hours off to play a round of golf. The first teed up, put a new ball on the tee, and drove a full 250 yards straight down the fairway. He stepped back underplaying his satisfaction and bowed to his partner. The partner reached into his bag, pulled out a brand new ball, set it on the tee, and with a terrific stroke, sliced the ball into the woods. Without looking around, he quietly reached into his bag, took out another new ball, teed up, then with a powerful stroke, hooked it off the course to the highway where it bounced among the cars and was lost. Again he reached for his bag, took out another new ball. This time he topped it causing it to dribble into the water hazzard. By this time his companion could remain silent no longer and said, "Why don't you use an old ball?" His partner said, "You know, to tell the truth, I've never had an old ball." Rust and moss are not our occupational hazards.

The presidents of most sizable corporations or colleges need a computer, and therefore, they need a computer director. You may see the president as the man most in need of retreading, although the newer generation of chief executives have had some briefing on the way up. For many of us this has been a second effort in a career already loaded with responsibilities and uncertainties. It is only recently that most managers have felt comfortable with management data. There has been a small identity crisis stemming from who is telling whom?

When I first arrived on the campus I now serve, I went around in the summer getting acquainted with the people I found in the offices and walkways. Some people became flustered if I dropped in on them unexpectedly, so I didn't always bother telling people who I was. The conversations were often more interesting that way. One day I saw a young man reading a book under a tree, and I walked over and greeted him. I asked him if he knew who I was. He took a long look and then said, "No, but if you remember your address I'll help you find your way home."

One of the strange inversions that confront us today is that a man's lifespan is now longer than the era in which a specific change takes place. Centuries ago, a woodcarver might never see the cathedral finished or the end of a hundred years war. Today the foreshorted time from idea to mass use makes it possible to live through the era of lighter than air craft or street cars. Only rarely does an idea have such strength that it takes longer than lifetime to see its full effect. Such is the case with the computer and a few other so-called million multipliers. Any idea that can multiply man's previous efforts by the factor of a million is likely to profoundly affect his way of approaching things. The social adaptation to an idea may occur in many ways over many years.

Most inventions and discoveries advance knowledge modestly and improve man's efforts by small steps like ten fold or occasionally a thousand fold. The wheel and the plough are in the ten fold category. The steam engine, dynamite, and the airplane are in the thousand fold category. Although gadgets don't tire as fast as people, the rate of doing something, like air travel to walking, is less than a thousand to one. It may be helpful to look at a few million multipliers that predate the computer and see what impact they have had. It is likely that more large-multiple inventions will follow. The categories are subjective, and you may find it fun to devise your own. The purpose here is simply to set up the analogy.

The turn of the century brought a flurry of excitement over electro-magnetism. By its study electrical messages in wires and later wireless by radiation gave man the ability to transmit data at the speed of light, a million times faster than previous methods. Morse code moved to television signals and the electomagnetic spectrum expanded to x-rays and lasers. Mass communication was further enriched by mass manufacturing techniques so that millions of people simultaneously watch a television program or converse on telephones. Halography has recreated three-dimensional reality and tremendous data storage capacity. There is a similar mass effect of millions of people driving cars or watching movies, but the original million multiplier effect is not there. The long term gain and net impact of the electronic revolution cannot be overstated. After more than seven decades it is still taking place.

Another million multiplier that barely preceded the computer is the thermonuclear reaction first developed as fission. In early years it was limited to violent explosions. The multiplier is clearly a million to one over other explosives, but on that multiplier hangs the development of ship propulsion, electrical power generation, as well as the balance of world peace. Yet to be harnessed is the power of fusion, which might well answer our need for an abundant and clean energy source. Although we may have seen the cresting of communication by electronics, the impact of the thermonuclear reaction is yet to be fully felt. Its potential lethal capacity is awesome. Yet, as with all other giants, man is likely to develop it and take his chances.

The computer takes a rightful place as the next million multiplier. Taking part of its clout from the electronics that give it nanosecond reaction time and also part from miniaturization and memory, it is a formidable entity exceeding mass manufacturing in speed and efficiency. Like other million multipliers, it was first felt to be magical and diabolical. Unlike the others, the monster has become the miniature hand-held pet much faster than the others. What are some of the effects it will still have, and how do they compare with what is happening to other million multipliers?

Inexorably, the toy of the elite becomes the gadget of the many. As the telephone operator disappeared as the intermediary for people using the telephone, so the language handler disappears in most of the transactions on the computer. The computer moves into homes, and like the telephone even to bathrooms. It becomes utilitarian, gets regulated, taxed and so becomes indispensable to the user. As we contemplate the third and a half generation of computer hardware, we should also contemplate the silent "last revolution" which puts the process irrevocably in the hands of the people. This is not to deny the continuing development of hardware even as satelites have replaced telephone poles; but the user has achieved a hands-on relationship which then determines the bulk useage. The demise of chauffers, elevator operators, and telephone operators is bleak evidence of the vulgarization of great technical breakthroughs. Not that this is bad. The public is likely to be the great beneficiary of major inventions and discoveries of our time.

In adapting these multipliers to people it is often necessary to write new languages that move a long way toward the comfort and understanding of the user. Telephone connecting is circumscribed today more by what numbers people can remember and use than what is technically most efficient. The same is becoming true of computer languages. In fact, I suppose languages will soon be known as people-languages, with the computer merely the intermediary. In turn, the capacity and versatility of the computer will change the way people input these devices. We don't, at least yet, sing the touch tone signals to the telephone, but since power-steering, we do drive with fingertips, and we are learning to feel comfortable in front of television cameras. The prospect of new languages inputs to computers not based on phonic data is promising as is the direct input of phonic data to computers.

Although the discovery of gravity does not qualify as a million multiplier, one of the insights gained from it has the great value of causing an inversion of contemporary thinking. The explanation of planetary motion and emerging idea of a solar system paved the way for developing the laws of acceleration and predicting the movements of other heavenly bodies. As we extrapolate to what may happen with computers, it becomes clear that inversions are likely. If we contemplate machines with cores of 40 billion sheets, each with four thousand characters, it becomes cheaper to store data in the computer than on paper. Not only do weather predictions become possible, but so do the relationships between the four major forces in the world....a subject Albert Einstein devoted the latter years of his life to. These forces of gravity, electromagnetism, the weak force, and the binding force of nuclei are different in strength by a factor of 10^{27}. The rarity of these experiences makes the computer the logical means of synthesizing data from billions of particle encounters. In still another direction the computer is likely to invert what we commonly thought were people characteristics. Now we depend on one person to recognize another or to reconstitute speech. Already we identify the authenticity of money by machine, even credit rating.... tomorrow, perhaps, the whole man.

The inventors of the Eniac (J. Presper Eckert and John Mauchley) at the University of Pennsylvania in 1941-42, were my classmates. When they labored with their great vacuum tube layout, they thought its great use would be to automate factories through systematizing technical processes. They did not forsee its complete takeover of the accounting and banking fields as its first great impact. But the movement is still young. It is simply moving where the greatest net gain can be obtained in human effort. So far this has been in calculations and reporting. Although further penetration and embellishments may be expected in this arena, the mainstream must still be the large clusters of work efforts from which man will be removed or elevated.

Another fellow classmate is Dr. Willis Ware of the corporate research staff of RAND. His writings on computers point to its prospects for the future. There are now 111,800 computers that perform collectively 40 billion additions each second of the year. The industry now expends 27 billion dollars a year and should, next year, pass the auto manufacturing industry in size. In 20 years the computer's speed has increased by 50,000; its size has decreased by 10,000; and the cost of a bit has decreased 110,000 times. The amazing thing is that the trend line for the seventies has continued as a straight line. Speed by the 80's is likely to exceed one billionth of a second. Density of components will rise to 4 million per square inch. Storage capacity is most promising. The Bell Telephone Laboratories can store 10 million bits in a three inch cube. The Department of Defense reports 10 trillion bits on a 4 x 6 inch film. The cost per bit in 1980 is likely to drop to .001 cents. The progress of a decade assures still another level of penetration into our social order. The greatest impact is yet to come.

The society we serve has a way of letting the reign of change fall on all. Not only will the users be modified in the development of powerful tools, but so will the developers. Witness how television production has changed the teaching and entertainment process. The star system has developed as have new roles for the production directors. The technician and the developer have been relegated to send class status. A comparable movement can be expected when ever a powerful tool is domesticated. The opinion makers, the politicians and the powerful users will make it their own.

The stakes are high. The electronic media have the capability of taking ideas to people, obviating the need for people to move to the resources. Already 90% of sports watching and movie viewing is done at home. The other part of the package is the return data from the individual to the collection point, for counting sorting and transactions of every sort. This is computer territory and a development that will be heralded as an energy saver and a great convenience. It will reduce countless frictions related to cars, conventions, summer resorts, and Christmas shopping. Whether it will be better for man to be thus independent in location and able to call up limited resources is in essence a philosophical question. It may lead to a new type of gregariousness. It will anoint the computer with ersatz emotions.

There is yet another million multiplier that could occur within our lifespan. Our probe of outer space for a friendly civilization could tell us what they know. Jupiter now seems more friendly. It could give us a leap forward in incredible ways. This is a topic best left to your moments of relaxation at a classical Greek symposium.

COMPUTER LANGUAGES OF THE FUTURE

David B. Ward
Staff Analyst
Honeywell Information Systems
Waltham, Massachusetts

The computer as an adjunct to man's mental capabilities differs from "other" machinery, in that man must "speak" more to computers! Computers (only binary computers will be addressed in our discussion) are constructed of highly restricted information components. To realize the high mechanical reliability of two state devices demands an enormous number of them to be of any benefit. It is this large number of simple components complexion that presents man with a difficult communication problem. Ironically the computer, itself, provides for the solution to these problems. The irony is that the more capacity we make available in computers, the more of it we must consume in communication, to utilize the capacity. Maybe there is a "computing limit" constant?

By what "language" do we speak to computers? It is a fact that all present, and foreseeable, computer's "understand" only strings of ones and zeroes! The numbers of these characters required to make computers of any use is beyond human consideration. One of the first tasks we set the computer to is the translation from more humanly usable languages to the (demanded) binary language.

First attempts to provide more human languages were still closely related to the computer componentry. "Assemby" languages provided mnemonic names for various registers and memory locations. These languages allow computer implementers, whose concern is not the solution of "user's" problems, to get the computer to do "anything at all". Generally their concern is far removed from the interest of most of the people who might like to use a computer. It might be noted that Charles Babbage (1826) in his improving and extending the Mechanical Notation (descriptive geometry), in order to communicate the description of his complex mechanical computers, invented probably the first "assembly" language. The "assembly" languages evolved into "problem oriented", "higher level" languages. Fortran provided for solving arithmetic equations, in a nomenclature much like engineers are accustomed to. A more elaborate language, Cobol, provided for solution of "business" problems. Algol gave more consideration as to the precise definition of its meaning. The ranks have swollen to a whole book full of languages, of which we are so well aware.

Formal artificial (computer) language investigations came as a by-product from attempting to use the computer to translate from one "natural" language to another. Chomsky's classification of languages provides a basis for comparing (especially computer) languages, as to their complexity of recognition. Automata and formal language theory provide a basis from which the automation of language translators can rise. It is in this approach that the advancement of future computer language will come.

There are two factors (in my mind) that most strongly influence the use of computer languages. Both of these factors are one of expense: monetary and psychological. If the use of a computer language is not financially beneficial (there is much that enters into this consideration) than the language has little hope of use. Compounding this consideration is the factor I am calling the psychological expense. The "apparent" psychological attractiveness of a language, to its users, effects the evaluation (consideration) of the use of a language, thus effecting the "apparent" economic considerations. As an employee of a prominent copier manufacturer I witnessed the use of "new" computers to emulate "old" computers. This was done upon the premise that "many" programs had already been written in a language that was only accommodated by the "old" computers. Consequently it was economical to (spend more) to maintain this "old" mode than to utilize what (eventially) was more economical, using the "new" computers in their more efficient "new" mode. One of the biggest factors (I contend from observation) was the psychological factor. The people that were responsible for the decision did not want to become involved with a new language that was different from one they were already used to! Using what you are familiar with is a predictable (psychological) factor effecting the utilization of computer languages. It obviously (justifiably) influences the economic consideration. The conclusion that there are better means of transportation than the automobile will not change our "habit" of using it.

Another, more realistic, economic consideration in the advancement of "better" language facilities is the cost of transferring existent implementations, in an "old" language, to a "new" language. This more an indication of an immaturity in the use of computers than an indication that there is an insurmountable problem. This is a result of the ill-definition (certainly non-formal definition) of our early languages. The arrival of the means to implement languages in a "better" way has arrived long after there were many languages in use. Given the "better" means of implementation provides (potentially) the means of automating their translation to "other" languages.

How about the efficiency of "high level" languages (in comparison to "assembly" languages). The bicycle is the most "efficient" form of transportation, but it is difficult to transport 300 people 3000 miles in 5 hours with a bicycle! There are computer operating systems efficiently implemented in assembly language, by 2000 efficient "ppogrammers, that do not work too efficiently at all. The whole problem of transferability, mentioned above, is largely a result (I maintain) of unfortunately implemented language translators - very efficient ones though! My conclusion is that we are not measuring "efficiency", but rather isolated factors effecting it.

How about the "tower of Babel" effect of alot of languages? Lets "standardize" our languages. The French government decides what the "standard" French is, what Frenchmen speak is something else again. The "standard" version of Cobol is atleast as many languages as the number of implementations of translators for it. Computer languages are much like natural languages. It is only in dictionaries and encyclopedias that we attempt to collect definitions of their complete meaning. We do not start with a complete specification. Languages grow in their use. At any one time we use a very restricted subset of any language. We even extend it for temporary "local" use. Some advocates of computer languages would like to defy this "gravity". It seems desirable to "fix" the meaning of a language once and for all. This is a result of the expensive translator implementation methods in use. What is needed is a language facility whose only purpose is to precisely define languages, in general. The languages defined might be quite changable. We will want to use these defining languages as input to computer programs to create translators, to lessen the cost of creating tanslators.

Languages that are to be utilized by people have some psychological requirements: they must be changable (humans do not like adjusting to new things and yet they crave change?), a language must not demand that vor have to be too precise (to do so enforces an unacceptable morality! Ok for the "other" person but I require more "flexibility" with which to express myself), etc. What I am striking at is that computer languages may be artificial, but human use of them is not. I will end my rather shallow examination of the phobales of computer language considerations and elaborate on the definition of "better".

Better implementation of what is better needed. "Structured programming" is the current catch word for better implementation. For language translators we can go one step further. Grammars, by which we might formally define the syntax of languages, correspond naturally to a "top down" definition. Let us use computers to automate translators. My work involves this effort and I appreciate the current limitations of the facilities provided (no good formal semantics). Never the less, much useful work has been accomplished (at universities) that we can take advantage of. Let we (the computer manufacturers) provide the tools for translator construction, rather than the translators themselves! We (Honeywell) will provide the "old chestnuts", Fortran and Cobol (even PL/I), but we will also provide a translator system as a customer product. Only the purchaser of computing equipment is in a position to know how he needs to communicate with his computer. The comuter manufacturer can not encompass this diversity of interest. The advent of micro-programmable hardware necessitates more general user translation facility, if the user is to benefit from micro-encoded machines. Computer manufacturers use micro-coding to implement their concept of what a computer should be. The inherent inefficiency of interpretive micro-hardware, in comparision with "hard-wired" hardware, necessitates attuning these machines to more specific use. The "general purpose" computer are more of financial benefit to computer manufacturers than to those who use them. I see the advent of "translator writing machines". These might take the form of a time-sharing service, the output of which is the definition of a specific micro-encoded machine. Hardware has now become software. The user must be provided with the facility to create his own (machine) definition. If you need your own computer, you need one precisely for your need. Otherwise your need is for a, rentable, computer service. How to distinguish which you need is a consideration of economics. The ($50) hand calculator is the current manifastation of the dedicated micro-hardward computer. Only volume manufacturer considerations prevent the creation of "one only" user defined micro-machines. Engineers have long been saying, "here's micro programming, what do you want us to *say* with it?". It is the ultimate user who must enter this conversation. Computer manufacturers are only an example of computer users. It is not their place to dictate the "nature" of computers. They can do so only on the basis of profitable capitaistic considerations, not in the best interest for mans benefit from computers.

Computer languages of the future should be originated by those who will use them. The "hand held" computers dedicated to a restricted function will have very specific languages. It will be practical to have "throw away" languages, languages invented for some fleeting need, whose life might be a day or a week. Speaking in "English", in the same sense as we restrictively use it for any specific conversation, is possible. The signifance of future languages is not what they will be, but the means by which we "implement" them. Hopefully both man and the computer will better understand what is being said.

TO COM OR NOT TO COM

Ernest L. Jones
Director
Data Systems and Services
Indiana University

TO COM OR NOT TO COM

Ernest L. Jones
Director
Indiana University Data Systems and Services

A few years ago some data processing professionals considered the producing of output of a computer on microfilm was almost sacrilege. Many people seemed to feel that this threatened their own positions as suppliers of information. At Indiana University and many other installations, throughout the country, COM has become an integral part of the data processing services that are provided our users.

How did we get there? What were the steps we used to determine whether COM was appropriate at Indiana University? What were the disadvantages? What were the advantages? Would it be a cost effective program? What did we stand to gain or lose?

In March 1972 we assigned the task of preparing a study for Indiana University on the use of Computer Output Microfilm to Mr. Ron Terrell, who was at that time a Lead Analyst and who is currently Assistant Director of Systems Programming and Operations.

To establish a level of understanding prior to proceeding, we should answer the question "What is COM?" I quote from the study presented by Mr. Terrell.[1]

> "COM is a method of converting data from a computer into human readable language on to microfilm. The result is a 17mm, 35mm, or a 105mm microform. In other words COM is a conversion process which allows going from magnetic tape to microfilm and omitting the hard copy [printed output] stage.

1. Computer Output Microfilm (COM) Study, Terrell, Ronald R. June 19, 1972, unpublished.

Two components are needed to create COM microform.

1. A properly formulated magnetic tape.

2. A COM unit which consists of a tape device, a film camera, and usually a cathode ray tube.

COM is intended to replace conventional paper output thus eliminating the purchasing and use of expensive carbon interleaved forms and the time necessary to decollate, burst and bind these forms. COM output onto film is roughly 20 to 30 times faster than a computer impact printer thereby saving expensive computer time.

A COM unit with a universal camera can produce microfiche as well as microfilm. COM-produced microfiche is 105mm film cut into six inch strips so that the end result is a 4 inch by 6 inch sheet of film. Each fiche contains 207 frames or pages of data, an index frame, and an eye-readable title, using a 42 to 1 reduction ratio.

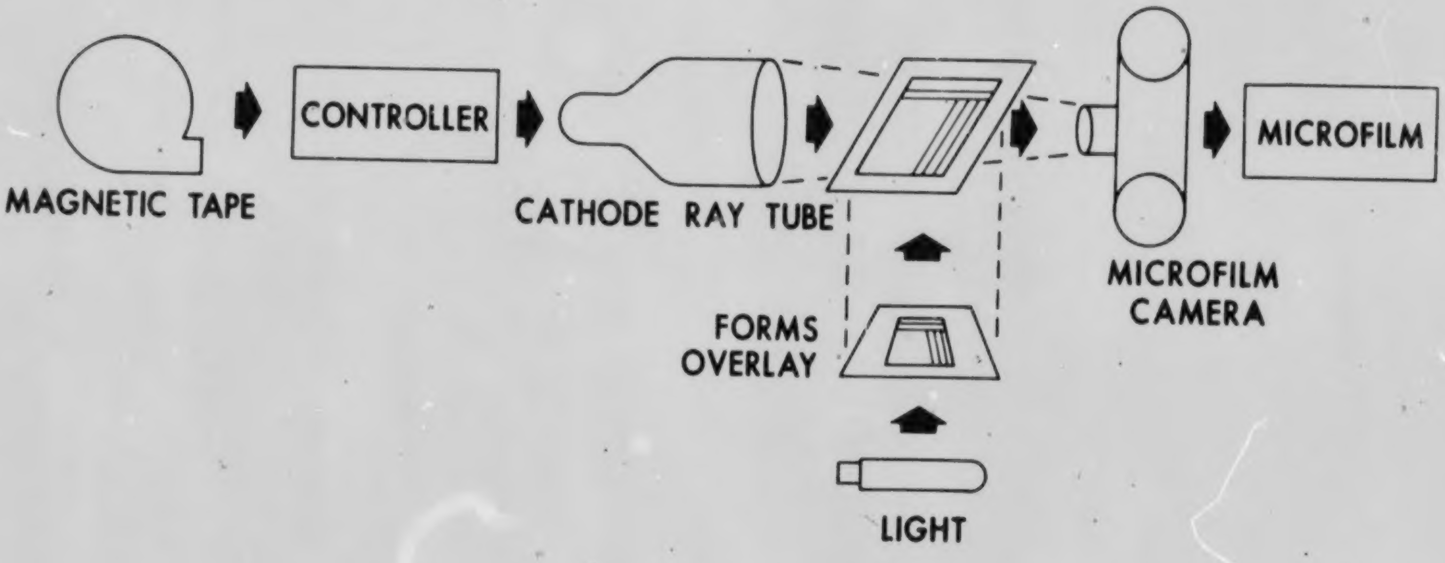
COM microfiche has several advantages over microfilm. A microfiche reader costs several times less than a motorized roll film reader. By using more than one reader several people can use the same fiche file simultaneously unless they need the same fiche. Also an individual document can usually be located faster on a fiche reader as compared to a roll reader."

In Illustration 1 below, the processing of the data by the computer has already been completed and we have obtained from that process a magnetic tape. This magnetic tape is then placed on the microfilm tape reader which when processed by the controller produces images of the output on a cathode ray tube. A microfilm camera then takes a picture of the cathode ray tube which may have superimposed on it a forms overlay. The result is a strip of microfilm. The complete process is shown in Illustration 2.

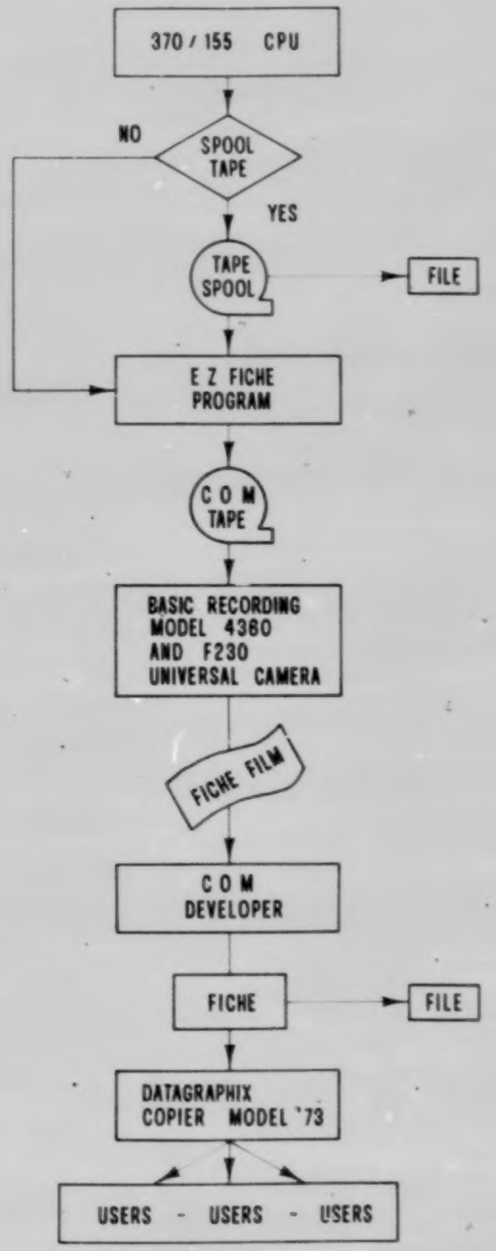
A comparison of the speed and mailing costs with paper output is presented in Illustration 3 and 4.

The first step in preparation for the study was a review of the literature in the area. Mr. Terrell then attended a four day seminar

COMPUTER – OUTPUT – MICROFILM RECORDING

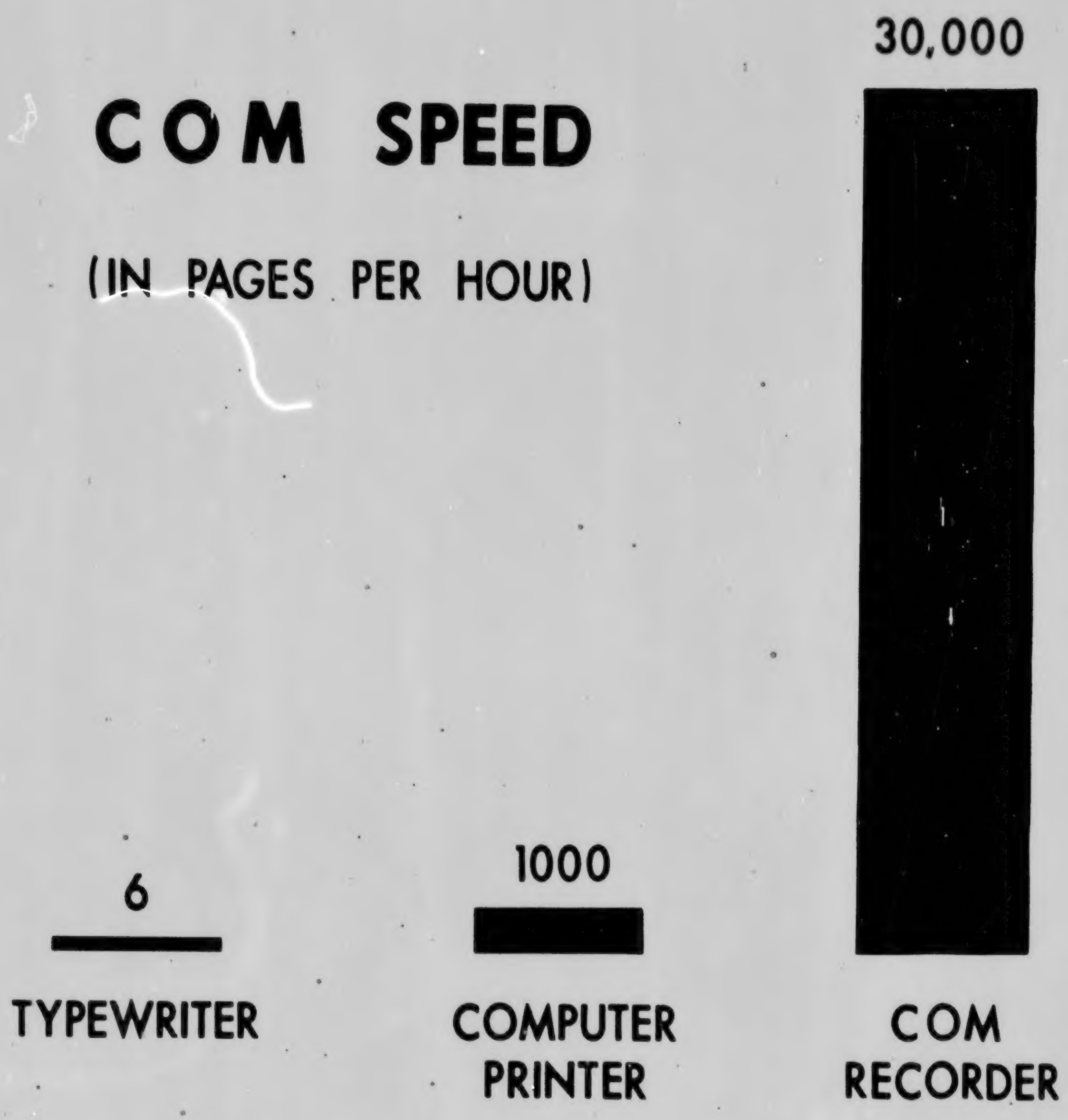

MAGNETIC TAPE
CONTROLLER
CATHODE RAY TUBE
FORMS OVERLAY
LIGHT
MICROFILM CAMERA
MICROFILM

CREATION OF MICROFICHE


370 / 155 CPU
NO
SPOOL TAPE
YES
TAPE SPOOL
FILE
E Z FICHE PROGRAM
C O M TAPE
BASIC RECORDING MODEL 4360 AND F230 UNIVERSAL CAMERA
FICHE FILM
C O M DEVELOPER
FICHE
FILE
DATAGRAPHIX COPIER MODEL '73
USERS - USERS - USERS

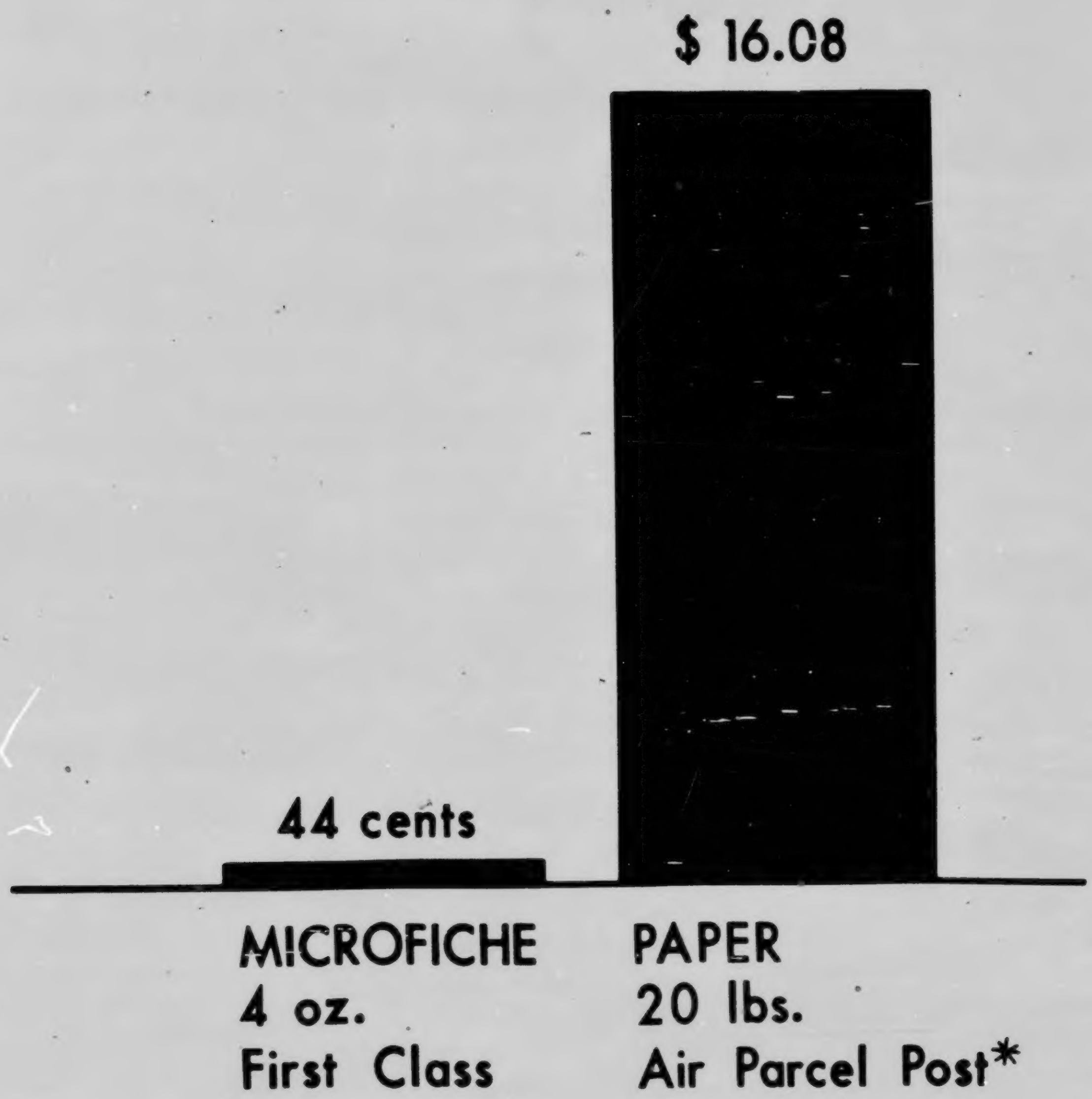
COM SPEED

(IN PAGES PER HOUR)



MAIL COSTS

Airmail, Coast to Coast



*Cost of 20 lb. paper report
Air First Class - $ 35.20

presented by Advanced Management Research on the subject. The program included recognized leaders in the industry who presented an excellent picture of the benefits to be gained and the problems to be overcome. A trip was also made to the University of Michigan Data Systems Center, who had installed COM equipment a few months before, to see what their experience had been and whether their experience was favorable.

The next two and a half months were spent doing a thorough study of all possible applications and investigating the impact on Data Systems and Services of a change in the method of processing the output from the computer.

Due to the method of funding for Data Systems and Services, a very serious constraint was placed on the justification for the changing of our procedures. Our department does not charge back to users except organizations associated with the University but not a part of the University, auxilliary enterprises and non-university users. Therefore, the cost of the equipment had to be borne by Data Systems and Services from current expense monies. This meant that the equipment had to be paid for through direct savings of actual dollars that normally would have been spent to produce the reports.

Since our department purchases all stock paper it was decided that the COM equipment had to be cost justified through paper savings alone. Many other factors were considered during the study such as savings in space, savings in postage, storage cabinets, computer time, and faster access to the data. These were investigated to determine the impact on the institution and to assist in convincing our users that this would be a usable product that they could accept.

The first step to determining whether COM was feasible at Indiana was to identify reports that were potential COM applications. The following guidelines were used:

1. Report must be repetitive and approximate 200 pages or longer.
2. Hard copy is generally not needed.
3. No annotation is required.
4. Report is internal to the University.

Information was gathered on all repetitive reports printed by the department and included the following information: Average number of pages per report; average number of lines per page; the number of copies printed; and the frequency of the report. A total of 48 reports were identified as potential COM candidates.

The University Registrar had already expressed an interest in converting some of his reports to COM so nine large volume reports were used as the basis for a cost analysis. Analysis indicated that the savings on paper alone would pay for almost two-thirds of the in-house lease cost for COM equipment, and if all reports were converted, the savings in forms cost on stock paper would amount to $800.00 per month more than the cost of the COM equipment and supplies. The study indicated that the break-even point would be about 100,000 original frames per month under the assumption that these would be printed on four-part stock paper at a cost of $15.73 per thousand sets. Therefore, the recommendation was to proceed with the installation of in-house COM equipment.

We did consider the use of a COM service bureau in Indianapolis but rejected it for the following reasons:

1. Travel of 50 miles each way for pickup and delivery.
2. Volume within one year would exceed the cost of in-house facility.
3. Software changes would be required if in-house equipment, when installed, was different than that used by the service bureau.

This approach is a viable one for an installation whose volume will not justify in-house COM equipment in the near future. At the time of our study 33,000 original frames and 1870 duplicate fiche would cost $748.00. The break even point was 87,000 original frames and 2,200 duplicate microfiche per month between in-house and service bureau rates.

Bids were solicited from three manufacturers of COM equipment and Stromberg Datagraphix was selected, based on the price and contractual arrangements available through that company. The equipment was installed in September 1972. The Purchasing Department invited producers of viewers to display their products at the Purchasing Department for view by potential user's of the COM equipment. The small Kodak Viewer was selected by the University Hospital in Indianapolis, and the other users selected the COM 150 produced by Micro Design Incorporated.

Don M. Avedon, technical director of the National Microfilm Association has listed a number of factors that have to be considered in selecting a COM recorder. He stressed that the importance of each factor has to be determined by the individual installation.[2]

'Character set and type style
'Cost
'Delivery schedule
'Equipment reliability
'Error detection capabilities

---

[2] "COM Helps Contain Paperwork Explosion," Avedon, Don M., Computerworld Volume VII, No. 44, October 31, 1973.

'Film size and format
'Film supply
'Forms overlay
'Image quality
'Job setup
'Magnetic tape format
'Manufacturer's reputation
'Operating environment requirements
'Plotting capability
'Programming and software
'Retrieval codes
'Service
'Throughput rate
'Transfer rate
'Type: alphanumeric or graphic

Once the decision was made the first development efforts were directed toward the nine student record reports which had the full support of the University Registrar and contact was made with all other departments whose reports had been identified as potential COM applications. As anticipated some of these potential users were less than enthusiastic and some are still "studying" the application of COM to their areas. Some reports were identified by the users which had not been included in our original survey and these were converted.

Illustration 5 is a comparison of the actual experience for the first twelve months of operation compared to the estimate taken from the original study. You will note that using the prices quoted in the study on cost of four-ply paper that the estimate of savings for the first year of operation was $515.00. Our actual savings was $1,715.46. However, paper cost has increased by twenty percent since the original estimate was made going from $15.73 per thousand to $18.91 per thousand. Since all of our cost for equipment and supplies is included in the total cost for the year our savings on the basis of the current cost of four-ply paper was $7,791.72 over cost including one third of the

COMPARISON OF COSTS

1st Year Operation of COM

	Pre-installation Estimate	Actual
Equipment & Supplies	$ 20,100.00	$ 19,704.62
Installation & Development, Amortized each of 3 years	2,685.00	2,685.00
Total Cost 1st Year	$ 22,785.00	$ 22,389.62
Savings on 4-ply paper, 14 7/8x11" @ $15.73/M	23,300.00	24,105.08
Net Savings	$ 515.00	$ 1,715.46

Savings on 4-ply paper at $18.91/M	$ 30,181.34
Installation & Operation of COM 1st Year	22,389.62
Savings at new paper cost	$ 7,791.72

AMORTIZATION FIRST YEAR

Equipment & Supplies	$ 19,704.62
Installation & Development	8,054.00
Total Cost 1st Year	$ 27,758.62
Savings on 4-ply paper at $18.91/M	30,181.34
Net savings after amortization	$ 2,422.72

installation and development cost. If you assume a full amortization of all costs during the first year we realize a net savings of $2,422.72. These are actual cash savings.

If we were billing our customers for our services and funds were actually changing hands we would have billed to COM users a total of $122,576.64 for printing the material on hard copy based on our current charge of 12¢ per hundred lines of print. Obviously there would have been a considerable savings to our users if they were in fact paying for the printing on hard copy. A month-by-month report of microfiche production is shown in Illustration 6.

One of the problems that obviously exists in our situation is the fact that we do not bill for our services except select customers such as the Indiana University Foundation, Auxilliary Accounts, and non-university users. Under the present operation there is little incentive for the user departments to be interested in the use of COM. They are happy with the hard copy output. To change to COM would mean disrupting their staff, and the cost of a viewer and perhaps a viewer/printer if they wish hard copy capability. The University Registrar was very interested in the use of COM to save space and expedite work since at the time we began investigating this system they were under some very severe space constraints. The remote campus registrars, who were even more crowded for space, were also anxious to change.

Some offices who in fact purchase their own custom forms have switched to COM because they realize that there will be a considerable savings to them particularly with the cost of paper increasing. The Student Loan office is one of those offices that did convert for that

COM PRODUCTION & EQUIVALENT COMPUTER PRINTOUT COSTS

Month	Production of Microfiche			4-Ply Paper Set Equivalents	Cost of Paper Equivalents	Computer Printer Cost Equivalent
	Originals	Duplicates	Frames (page equivalents)			
October '72	120	1652	330,400	82,600	$ 1,561.97	$ 6,343.68
November	126	1489	297,800	74,450	1,407.85	5,717.76
December	223	2401	480,200	120,050	2,270.15	2,219.94
January '73	194	1079	215,800	53,950	1,020.19	4,143.36
February	160	1094	218,800	54,700	1,034.38	4,200.96
March	236	2545	509,000	127,250	2,406.30	9,772.80
April	376	3628	725,600	181,400	3,430.29	13,931.52
May	290	2196	639,200	159,800	3,021.82	12,272.64
June	263	1184	236,800	59,200	1,119.47	4,546.56
July	868	7494	1,498,800	374,700	7,085.58	28,776.96
August	515	3787	757,400	189,350	3,580.61	14,542.08
September	729	2372	474,400	118,600	2,242.73	9,108.48
TOTALS	4100	30,921	6,384,200	1,596,050	$ 30,181.34	$ 122,576.64

reason and they saved the cost of the viewers and netted about eight to nine hundred dollars the first year and will have a net savings of approximately $2,400.00 for the present fiscal year. The Indiana University Foundation has just recently converted all reports to COM. They too, expect to make a considerable savings in forms cost, as well as a savings from our department on the basis of printed lines since they are one of our paying customers.

Since the guidelines for the study were set to show a hard dollar savings for our department as a justification for the installation of the COM equipment, the entire report was based on savings on the four-part paper alone. However, there are other savings, which I've alluded to from time to time, which can accrue not only to ourselves but to our customers.

1. Savings in space in the storage of microfiche as opposed to hard copy.
2. Efficiency of handling.
3. Savings of postage where reports were being shipped to our remote campuses.
4. Savings in personnel time from preparing those reports for mailing.
5. Savings in our department at the printer since we do not have to change paper frequently as would be the case in hard copy print-out.
6. Savings in bursting and deleaving.

Space savings are dramatic. 75,000 pages of ordinary computer print-out weighs 500 pounds, requires 10 cubic feet of file space and costs about 1/4 of a cent per page while the same output on microfiche weighs only nine pounds, occupies .018 of a cubic foot of space and costs one tenth of a cent per page.

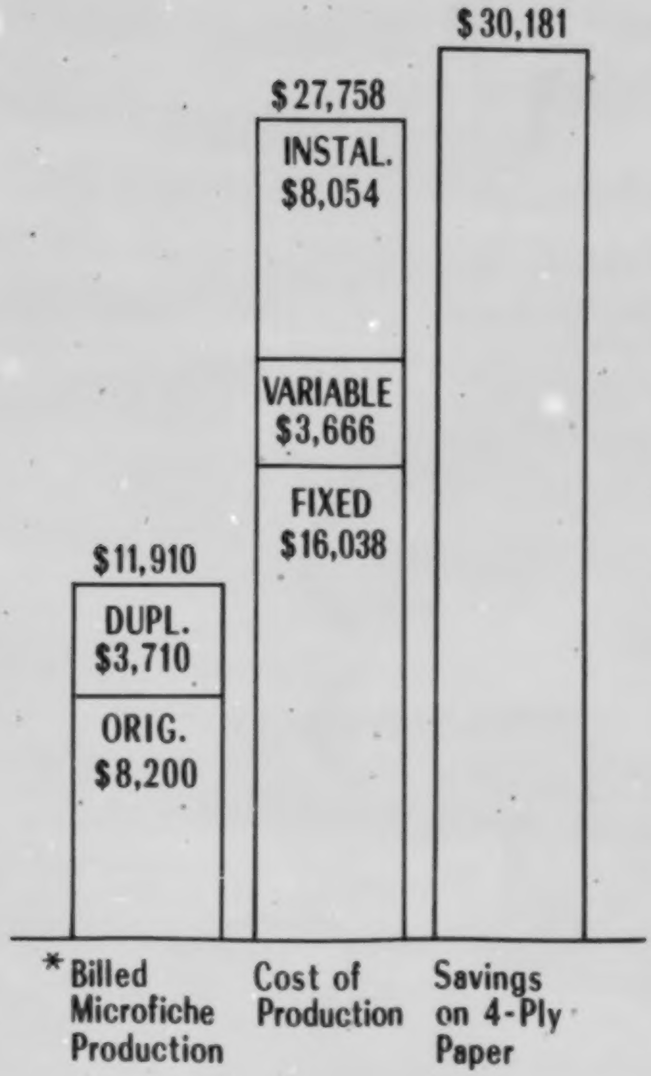
In the report we did show that this change in output mode would not increase our labor cost. We did an examination of the time required to deleave and burst multiple-ply sets and compared that with the time it would take to produce the original, and duplicate microfiche, for the reports converted to that medium. The study estimated that bursting and deleaving 10,566,000 pages of material would require 1,800 hours per year in our bursting and deleaving section. If the same pages were produced by the COM equipment then we estimated six hours per day or approximately 1,500 hours per year or a savings of 300 man hours each year. This would represent approximately $1,800.00 per year. These estimates proved to be quite reliable, but we have recently changed our duplicator from a manual feed to an automatic feed which reduced the required time by two hours per day. This will account for another 500 man hours per year or an additional $3,000.00.

The recording equipment and the developer have been extremely reliable. The only difficulty we have encountered with the equipment was the duplicator. The first duplicator required the insertion of the original fiche and film into the duplicator and then passing the copy through the machine twice to provide a good quality output. However, the quality was still difficult to control resulting in overdeveloped, and underdeveloped copies. Film wastage was high thus increasing our costs. It was often necessary to re-do a single fiche several times in order to get the quality copy desirable for the customer. A new duplicator, which was installed in October 1973, is very similar to a Xerox machine. You insert the original fiche in a frame, turn the frame under the mechanism and set the number of copies you wish. Press the start button and the machine automatically produces the correct number of copies. While it is producing that set of fiche you may insert a second original and be prepared immediately to produce copies of the next microfiche. The quality from this machine has been extremely good and very consistent and there is virtually no wastage in film. The film comes in a roll which is automatically cut by the machine as it produces the duplicate fiche.

Our present rates for original and duplicate fiche are not sufficient to recover the cost of the equipment and supplies as indicated in Illustration 7, but since the original goal was to offset the cost of the equipment by savings on four-ply paper we have not been overly concerned. From the figures we have accumulated over the past year our rate should be $2.25 per original fiche instead of $2.00 and 15¢ per duplicate fiche rather than 12¢. Using these figures and based on our current production level we could offset our costs without considering savings on four-ply paper. Since the majority of our customers do not pay it is doubtful we will change the rates but will review them once again before the next fiscal year begins.

While our progress is slower than we would have liked, nevertheless it has been according to our estimates. The word is beginning to spread, however, and we anticipate that there will be more and more users who will decide that COM is the way to go rather than cluttering their offices with masses of paper. Our remote campuses are beginning to use their remote job entry terminals to produce COM output rather than printing on their local printers and are encouraged to do so by some of

COM COST COMPARISON
1st year graphically



* Although statements are prepared for all services rendered, most University users are not expected to reimburse DS&S for the cost thereof.

their users in order to conserve space in their offices.

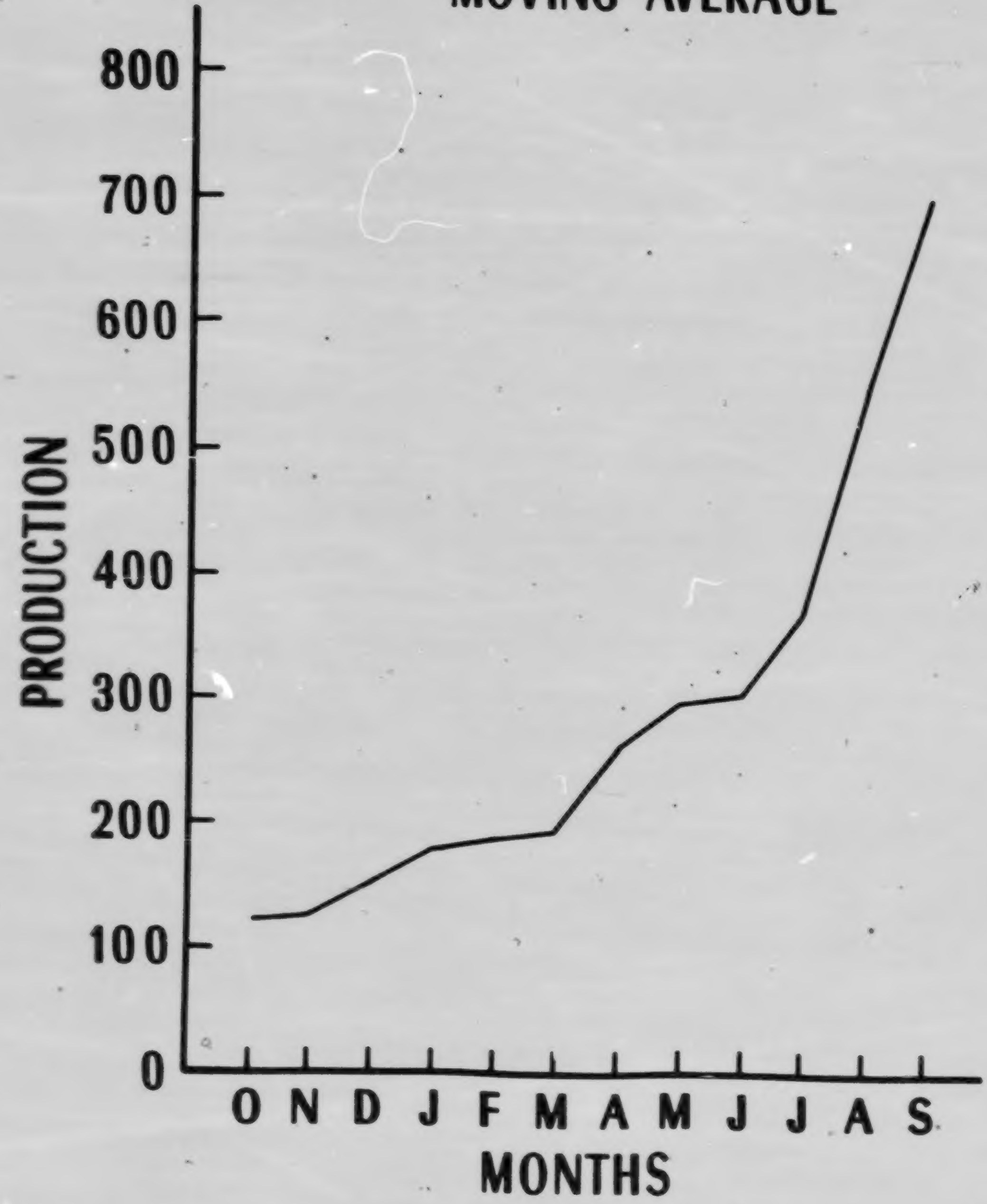
We have not converted our student permanent records to this system but have been exploring this possibility. At the present the Registrar is investigating the cost of automatic fiche retrieval equipment and we are investigating the software to provide an on-line index which can be accessed from their cathode ray tube terminals rather than having index on fiche themselves. This application will save a great deal of floor space for the Registrar and will eliminate a great deal of personal handling of the records.

Illustration 8 shows the growth in the production of original fiche.

If you are currently billing your customers for services, you may find extremely willing converts to COM.

To COM or not to COM, that is a QUESTION?

COM PRODUCTION OF ORIGINALS
MOVING AVERAGE



CAUSE DBMS PROJECT REPORT

Paul J. Plourde
Director
Computer Center
Amherst College

Bard F. White
Director
Management Systems
University of Massachusetts

COLLEGE AND UNIVERSITY SYSTEMS EXCHANGE

DATA BASE MANAGEMENT SYSTEMS (DBMS) PROJECT

Interim Report

The CAUSE Data Base Management Systems Project was established in November 1971 to provide information to member institutions about the availability and use of data base management systems (DBMS) in higher education.

There has been increased fan-fare concerning DBMS systems during the past several years. There are an increasing number of colleges and universities who are either in the process of evaluating, or using, a DBMS.

There has been, and continues to be, much disagreement about what such systems can and cannot do and how one should go about evaluating or implementing such a system. Even with this disagreement, it remains clear that the evaluation and implementation process of a DBMS can be an enterprise equally as involved and costly as installing new hardware.

Since one of the main problems of decision makers in higher education is the availability of meaningful, well-organized, timely data, a major task of the administrative systems staff becomes one of how to organize data in a meaningful fashion for rapid retrieval.

In the past, the most prominent solution to this problem of making timely information available to administrators was to install a computer and begin the computerization of routine functions such as payroll, accounting, course registration and student reporting. The resultant systems usually were characterized by the development of various separate files for each application.

Institutions who have had access to computers for several years have invested many man-years of effort in developing these individual systems. One of the inherent problems with these stand-alone systems is that meaningful interrelationships between data in the several systems are seldom drawn. Furthermore, attempts to develop reports which cross these application systems, or sub-systems are seldom successful. In an attempt to over-come this problem, and to assist in the solution of the information crisis, an increasing number of institutions have attempted to develop information systems that revolve around an integrated data base. The purposes of information systems designed in this manner are to: (2) simplify retrieval of information by users, (b) draw meaningful interrelationships between data and multiple information centers, (c) maintain and update information in the data base and (d) provide the capability of easily defining and modifying the required data base.

47

It should be noted that this approach to information processing is a radical departure from the traditional approach of capturing and reporting data in individual sub-systems; but it is necessary, since the existence of files tied to individual sub-systems does not properly reflect the true workings of an institution. Data does not exist as discrete information in each office for the exclusive use of that particular office. Quite the contrary, data exists in information centers as potentially assessable by anyone in the institution who has clearance to access that particular data. More important, perhaps, is the question of bringing together data that exists in individual sub-systems and drawing meaningful interrelationships between two or more pieces of information.

In view of the fact that people are diverting from the traditional approach of information processing, and that much time and effort has been expended in a number of institutions in gathering information about data base management systems, it was felt that CAUSE could serve a useful purpose by establishing a project that would disseminate information about the state-of-the-art as it related to such systems in higher education.

The first task of the project members was to define what constitutes a DBMS. To arrive at a definition, the group drew upon previous working definitions used by CODASYL and the Mitre Corporation and derived the following definition for DBMS:

> A DBMS is a generalized software package that is user oriented and can perform tasks related to file definition, creation and update, and query and reporting. More specifically, a DBMS is a software package that provides the following facilities: (1) a logical structure for presenting data, (2) a procedure for maintaining and accessing the data in a physical storage device, (3) associative structures relating the data elements within a data base, and (4) a set of rules and conventions for effecting actions against data base. This software is application dependent and it can process personnel, student, course, facility, or financial data bases with equal facility without any modification to either the operating system or application programs.

Having decided upon a definition, the next item on the agenda was to establish objectives for the project. These are listed below along with the method of gathering data for each step in the project and the current status of each of these items. You will note that the ultimate objective of the project is the publication of a CAUSE Monograph in early summer of 1974 that will be made available to all members. The contents of this monograph are indicated in the list of objectives and a preliminary version of the bibliography is attached to this paper as well as a list of the members of the DBMS Project.

CAUSE DATA BASE MANAGEMENT SYSTEMS (DBMS) PROJECT

Objectives	Method	Status
To identify through the use of a survey instrument those educational institutions that might utilize or are currently using a data base management system	Survey	Completed
To identify and establish a list of those organizations, both private and public, which will supply software that may be defined as a data base management systems	Form Letters	Completed
To solicity from educational institutions their comments on the system they are using, the criteria they used in selecting a particular DBMS, and make that information available through CAUSE	Survey	Completed
To establish a working group within CAUSE whose prime objective is the evaluation and establishment of DBMS criteria and ultimate publication of a DBMS document	CAUSE Newsletter	Completed
To compile, analyze, and summarize the data gained through the use of the CAUSE survey, contributions from the DBMS working group, and contributed papers from invited member institutions		In Progress
Publish and make available through CAUSE a monograph containing chapers on:	Monograph	Second quarter 74 publication

1. Historical perspective of DBMS development, definition of a total system, advantages and disadvantages, and expectations
2. Implementation guidelines including conversion, backup, security, integrity, testing and training
3. Functions of a data base administrator
4. Results of DELPHI study - evaluation instrument
5. DBMS abstracts - vendor responses, list of educational users and their critical comments
6. Bibliography of DBMS articles and publications

Task Force

Michael Abbott
Director
Management Information Systems
Texas Coordinating Board

Harry Brooks
Director
Operations Research
University of Vermont

Glenn Harvey
Director
Computer Services
DePaul University

Wallace C. Knapp
Director
Systems Development
American University

William McKelvey
Director
Systems Analysis
Ohio State University

Paul J. Plourde
Director
Three College Computer Center
Amherst College

Kenneth E. Shostack
Staff Analyst
Information Technology
Harvard University

Bard F. White
Director
Management Systems
University of Massachusetts

DATA BASE MANAGEMENT SYSTEMS

BIBLIOGRAPHY

1. Meadow, C. T. *Analysis of Information Systems*. New York: John Wiley and Sons, 1967, pp. 151-252.

2. Schubert,R.F. "Basic Concepts in Data Base Management Systems," *Datamation*, July 1972, p. 42.

3. McLaughlin, Richard A. (ed.). "Building a Data Base," *Datamation*, July 1972,p.51.

4. Briggs, Peter L. "Buying Packaged Data Management Seems Best," *Computerworld*, pp. 11-12.

5. Pan, George S. "Characterization of Data Management Systems," *Data Management*, June 1971.

6. White, Bard F. "Comparative Evaluation of Data Management Systems," Version 2.0, (Monograph) Durham, N.C.: NLHE, April 10, 1973.

7. Learman, I., *Cost Benefits Analysis in the Design and Evaluation of Information Systems*, Fall Joint Computer Conference, 1971.

8. Gruenberger, Fred (ed.). *Critical Factors in Data Management*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1969.

9. Elwell, Harry H. "Data and Information Systems Management," *Management Services*, November-December 1967, pp. 40-51.

10. ____________. *Data Base Administration Survey Report*. Columbus, Ohio: Ohio State University, Office of Administrative Systems, July 1972.

11. Minter, John and Ben Lawrence (eds.). "Data Base Approach to a Management Information System," *MIS: Their Development and Use In Administration of Higher Education*; proceedings of a seminar held in Washington, D.C., April 1969. Available from ERIC, ED042427, p. 114.

12. Sharp, Robert. "Data Base Bibliography," Association of American Medical Colleges, May 10, 1971.

13. Cohen, Leo J. "Data Base Considerations and Implementation Techniques," *Data Management*, September 1972.

14. Metaxides, A. "Data Base Management," Newsletter of the ACM Special Interest Committee on Business Data Processing, Attachment 6, Association of Computing Machinery, New York, December 1967.

15. Sharp, Robert. "Data Base for an Integrated Information System," Association of American Medical Colleges, May 10, 1971.

16. Plourde, Paul J. "Data Base Management and the Design of a Total System for a College or University," Amherst College, January 14, 1972.

17. ____________. "Data Base Management Study - Administrative Data Processing," University of Utah, January 30, 1973.

18. Leavitt, Don. "Data Base Users Must Face Hazards," (Staff) *Computerworld*, July 11, 1973.

19. _______________. "Data Bases: There Really is a Better Way to Manage Your Files," ADL Systems, Arthur D. Little, June 1, 1973.

20. Romberg, Bernhard W. "Data Bases: There Really is a Better Way to Manage Your Files," *Infosystems*, June 1973.

21. McIntosh, Stuart and David Griffel. "Data Management for a Penny a Byte," *Computer Decisions*, May 1973.

22. _______________. *Data Management Systems Catalog* (MTP-139). The Mitre Corporation, January 1973.

23. _______________. *Data Management Systems Survey*. The Mitre Corporation, May 1969.

24. D'Imperio, M. "Data Structures and their Representation in Storage," Annual Review in Automatic Programming, Volume 6, Long Island, N.Y.: Pergamon Press, 1969.

25. _______________. "Debate on Data Management," *EDP Analyzer*, Vol. 10, No. 3, March 1972.

26. Dodd, George G. "Elements of Data Management Systems," *Computing Surveys*, June 1969.

27. Traver, Cheryl M. "Evaluation of File - Management Software Packages," (Systems Specialist) Project Info, 1969.

28. _______________. "Feature Analysis of Generalized Data Base Management Systems," *Codasyl*, May 1971.

29. Steig, Donald B. "File Management Systems Revisited," *Datamation*, October 1972.

30. Lefkovitz, David. *File Structures for On-Line Systems*. Washington: Spartan Books, 1969.

31. Fry, James P. "Generalized Data Base Management Systems: A Key Step Toward Solving the Data Bank Problem," University of Michigan, ISDOS Research Project, February 1971.

32. Fry, James P. "Generalized Data Base Management Systems: An Introduction," University of Michigan, Working Paper, September 15, 1970.

33. Gilb, Tom. "Generalized Data Base Packages Not Worth Cost/Risk," *Computerworld*, May 16, 1973.

34. _________________. "Generalized Data Management Systems: Sane Perspective," University of Maryland Computer Science Center, 1970.

35. ______________. "Generalized Information System Application Description," IBM Manual No. H20-0179, White Plains, N.Y.: International Business Machines Corporation.

36. McGee, W. C. "Generalized File Processing," Annual Review in Automatic Programming, Volume 5, Long Island City, N.Y.: Pergamon Press, 1968.

37. Borkovite, P. E. "Generalized Software Often Justified," May 30, 1973.

38. Harvey, Glenn B. "Getting to First Base with Data Base," DePaul University-Cumrec Conference, May 2, 1973.

39. ______________. "Introduction to Feature Analysis of Generalized Data Base Management Systems," ACM Communications, May 1971, pp. 308-318.

40. Kurz, R. C. "Long Live the Data Administrator," Datamation, March 1973.

41. Olle, T. William. "MIS: Data Bases," Datamation, November 1970.

42. Madnick, S. E. and J. W. Alsop II. Modular Approach to File System Design. Montvale, N.J.: AFIPS Press, 1969, pp. 1-13.

43. Deardon, John. "Myth of Real-Time Management Information," Harvard Business Review, May-June 1966, pp. 123-132.

44. ______________. "Organization Retrieval and Sorting of Information in a Digital Computer," SDC Field Note FN-1384, S-1, System Development Corporation, Santa Monica, California, 1959.

45. Shively, William C. "Packages Software vs. In-House Programming," Data Management, September 1971.

46. ______________. "Proceedings of Second Symposium on Computer Center Data Base Systems," SDC Technical Memorandum No. TM 2624/100/00, System Development Corporation, Santa Monica, California, 1965.

47. Welke, Larry. "Review of File Management Systems," Datamation, October 1972.

48. ______________. "Review of the Codasyl Report," Codasyl, August 1971.

49. Prendergast, Lawrence S. "Selecting a Data Management System," Computer Decisions, August 1972, pp. 12-15.

50. Curtis, Gilbert and Kenneth Falor. "Solve your Software Headaches with a Generalized Program," Computer Decisions, February 1971, pp. 34-38.

51. Fry and Gosden. "Survey of Management Information Systems and Their Languages," Mitre Corporation, May 1968.

52. Madill, J.P. and J.K. Kuss. "Techniques for Evaluation of Generalized Data Management Systems," Simon Fraser University, CUMRC, May 1972.

53. Kriebel, Charles H. "Techniques of System Design and Development," Data Base Magazine, Summer 1970, pp. 19-25.

54. __________. "The Corporate Data File," EDP Analyzer, November 1966.

55. Vorhaus, A. and R. Wills. "Time-Shared Data Management System: A New Approach for Data Management," SDC Technical Memo No. SP-2747, System Development Corporation, Santa Monica, California, 1968.

56. __________. "Trends in Data Management, Part 1," EDP Analyzer, May 1971, Vol. 9, No. 5.

57. __________. "Trends in Data Management, Part 2," EDP Analyzer, June 1971, Vol. 9, No. 6.

58. __________. "Using Shared Data Files," EDP Analyzer, November 1970.

ADDENDUM

1. Lyon, John K. An Introduction to Data Base Design. New York: John Wiley, 1971.

2. Patterson, Albert C. "Data Base Hazards," Datamation, Vol. 18, No. 7, July 1972, p. 48.

3. Aron, J. D. "Information Systems in Perspective," Computing Surveys, Vol. 1, No. 4, December 1969, p. 213.

4. Traver, Cheryl M. "Policy Issues Related to Data Base Management," Address to Second National Forum on Higher Education Management, November 1973, 10 pages.

5. Wendler, C.C. Total Systems Characteristics and Implementation. Cleveland, Ohio: Systems and Procedures Association, 1966.

DBMS DELPHI Study

Bard F. White
Director
Management Systems
University of Massachusetts

DATA BASE MANAGEMENT SYSTEM DELPHI STUDY

College and University administrators face a bewildering array of options in choosing computer hardware and software to support their institutional goals and objectives.

In an effort to aid administrators in selecting the computer software that best fits the needs of their institution, the author began some time ago developing an instrument to evaluate data base management systems. Pilot tests of this instrument indicated that weights for each evaluation question should be added. It was also determined that the best way to arrive at the appropriate weights would be through the use of a Delphi technique involving a number of practitioners in the field who had experience with data base management systems.

The questions were taken from the data management system evaluation instrument developed by this author and a Delphi instrument was developed to establish weights.

The Delphi technique consists of having experts in the field complete a series of questionnaires interspersed with controlled feedback on the responses of the other participants. This method provides anonymity to the participants, thus minimizing the influence of personal and political interests on their decisions.

The Delphi technique consists of four steps:

1. First, each participant is asked to mark his opinion on a specific topic.

2. Second, each participant is asked to evaluate all of the opinions in terms of a given criterion.
3. Third, each participant receives a list summary of all responses, and if his view differs from the most frequent responses, he is asked either to revise his opinion or to indicate his reason for not doing so.
4. Finally, each participant receives the list with an updated summary including minority opinions, and is asked to repeat or revise his own opinion.

The primary value of the Delphi technique is that it preserves the virtue of independent thought, and simultaneously permits the participant to tap the knowledge of others. In other words, the technique assures that opinions arrived at independently will be informed opinions, since each respondent is advised of the responses of others in summary form.

The following six pages are examples of the Delphi Instrument employed to establish weights on DBMS evaluation criteria. Two cycles of the Delphi have been completed. The third and last cycle of the Delphi is scheduled for April of 1974 with publication of the results in July of 1974.

DMS EVALUATION INVENTORY

Instructions:

Data Management Systems serve a number of objectives, however, in order for a system to be classified as a data management system, it must meet certain criteria. What do you consider to be the important aspects of a data management system? External to the actual operation of the software, what are other objectives the system must satisfy in order to achieve an institutional goal?

Each statement in this questionnaire is presented once, but you will be asked to react in two different ways:

First: How important __is__ the function to what you believe a data management system is?

Second: In your judgment, how important __should__ the function __be__ in order to achieve an objective at your institution?

__EXAMPLE:__

EVALUATION QUESTION		OF EXTREMELY HIGH IMPORTANCE	OF HIGH IMPORTANCE	OF MEDIUM IMPORTANCE	OF LOW IMPORTANCE	OF NO IMPORTANCE
1. What is the Maximum number of files the system can process or define?	IS	--- ( )	(✓)	( )	( )	( )
	SHOULD BE	--- ( )	( )	( )	(✓)	( )
2. What is the total number of institutions using the package?	IS	— ( )	( )	( )	(✓)	( )
	SHOULD BE	— ( )	(✓)	( )	( )	( )

In the first example above; the person has indicated that he believes the question "What is the maximum number of files the system can process or define?" is of high importance to the definition of a data management system, but that it should be of low importance to an institutional objective. On the other hand, example #2 is viewed as not being a critical factor in determining whether the software is indeed a DMS but the question is of high importance in addressing the institutional objective of a system's credibility, stability, and acceptance.

This questionnaire is completely confidential. No one will see your responses except professional staff working on this project. All results will be summarized by groups; individual results will not be released. However, for correspondence and return of questionnaires, we need your name and position. Please complete the following:

Name: ______________________________

Position: ______________________________

Name of Institution: ______________________________

Mailing Address: ______________________________

____

Is your institution (mark one) Public: ☐ or Private: ☐

Student FTE's:	Under 10,000	10,000-20,000	Over 20,000
(mark one)	☐	☐	☐

Type of computer you are now utilizing? ______________________

Note:

Participants in this study will receive the summarized results of the Questionnaire plus a copy of the final evaluation instrument.

EVALUATION QUESTION		OF EXTREMELY HIGH IMPORTANCE	OF HIGH IMPORTANCE	OF MEDIUM IMPORTANCE	OF LOW IMPORTANCE	OF NO IMPORTANCE
A. COST						
1. What is the direct cost (purchase price) of this package?	IS SHOULD BE	( ) ( )	( ) ( )	( ) ( )	( ) ( )	( ) ( )
2. Is the package available on a lease basis?		( ) ( )	( ) ( )	( ) ( )	( ) ( )	( ) ( )
3. If the answer to question 2 is yes, what is the lease price?		( ) ( )	( ) ( )	( ) ( )	( ) ( )	( ) ( )
4. Are the options available (above) suitable for us to budget for at this time and into the future?		( ) ( )	( ) ( )	( ) ( )	( ) ( )	( ) ( )
5. Is it possible to acquire this package in modules so that we need only acquire what we need at this time, yet add modules at a later date?		( ) ( )	( ) ( )	( ) ( )	( ) ( )	( ) ( )
6. Can any of our lease costs be used toward possible subsequent purchase of the package?		( ) ( )	( ) ( )	( ) ( )	( ) ( )	( ) ( )
7. Upon purchase or lease what are the estimated indirect (hidden) costs to the user for?		( ) ( )	( ) ( )	( ) ( )	( ) ( )	( ) ( )
a. known modifications before installation on our machine.		( ) ( )	( ) ( )	( ) ( )	( ) ( )	( ) ( )

EVALUATION QUESTION	OF EXTREMELY HIGH IMPORTANCE	OF HIGH IMPORTANCE	OF MEDIUM IMPORTANCE	OF LOW IMPORTANCE	OF NO IMPORTANCE	COMMENT
A. COST						
GROUP RESP.			1.9			
1. What is the direct cost (purchase price) of this package? YOUR FIRST RESP.	( )	( )	(X)	( )	( )	
YOUR SECOND RESP.	( )	( )	( )	( )	( )	
2. Is the package available on a lease basis?	( )	( )	1.7 (X)	( )	( )	
	( )	( )	( )	( )	( )	
3. If the answer to question 2 is yes, what is the lease price?	( )	( )	2.0 (X)	( )	( )	
	( )	( )	( )	( )	( )	
4. Are the options available (above) suitable for us to budget for at this time and into the future?	( )	( )	2.0 (X)	( )	( )	
	( )	( )	( )	( )	( )	
5. Is it possible to acquire this package in modules so that we need only acquire what we need at this time, yet add modules at a later date?	(X)	2.4 ( )	( )	( )	( )	
	( )	( )	( )	( )	( )	
6. Can any of our lease costs be used toward possible subsequent purchase of the package?	( )	2.9 ( )	( )	( )	(X)	
	( )	( )	( )	( )	( )	
7. Upon purchase or lease what are the estimated indirect (hidden) costs to the user for?						
a. known modifications before installation on our machine.	( )	(X)	2.0 ( )	( )	( )	
	( )	( )	( )	( )	( )	

Upon completion of the first round of the Delphi study, the six most important questions in evaluating a data base management system were as follows:

1. Is the package operational now, and has it been in a user environment for several years?
2. Are there institutions similar to ours using the package now?
3. Can the package add, delete, and modify the data base simultaneous with query and report generation?
4. Can appropriate backup copies of the data base including the latest transactions, be generated in case a file is destroyed?
5. Is backup/restart available?
6. Can the package retrieve data from two or more files within the same run?

On the other end of the scale, the four least important questions were as follows:

1. Are operators available for sine, cosine, tangent?
2. Is there a non-disclosure agreement in the contract?
3. May the system be made available to other users within the organization?
4. Can ALGOL, APL or JOVIAL be called during execution?

The question concerning cost or purchase price of the system - "What is the direct cost of the package?" ranked 210 out of 248 questions.

The final results of the Delphi study will be made available through the CAUSE Data Base Management Systems Project at a later date.

In addition to the Delphi study, the author is also developing an instrument to facilitate the comparative analysis of one data base management system against another. Upon completion, the publication will be made available through the CAUSE Document Library.

This instrument, when completed, should help guide the evaluation process when selecting a data base management system. Specifically, it will be of assistance in identifying institutional goals, determining measurable objectives, establishing constraints, and then determining if a particular system will be appropriate for the institution.

Sections of the instrument will deal with costs, availability, hardware requirements, software environment, quality, documentation, support, and contracts.

When package users are asked what they would do differently if they had it to do over again, and what advice they would offer to others, the answers usually fall into the following categories:

(1) They would check more carefully into the modifications required;

(2) they would demand more on-site support and training in operations and maintenance;

(3) they would require more thorough testing; and

(4) they would examine the documentation more thoroughly, and would check with other users.

This instrument will outline what a prospective user should look for when evaluating a data base management system so he may avoid many of the pitfalls already experienced by others.

The following six pages are examples of forms from the comparative evaluation instrument and include forms for goal statements, a task list, an example of comparison statements with options for weights, and a summarizing form.

Questions about this comparative evaluation instrument should be addressed to the author.

GOAL STATEMENT: Computer Center -

To develop and implement a management information system that satisfies the day-to-day information processing needs of operating level management and the short and long range needs of policy and top-level decision-making management

Developed By B. White

OBJECTIVES:

1. To provide the user with interactive, man-machine capability in user intelligible form.

2. To utilize a data management tool that minimizes the burden of application programming; this tool should permit for flexible use in taking the data base approach to systems planning.

3. To maximize the development potential of current staff.

4. To provide a continuous flow of control data and to provide a high degree of data security and total -- both hardware and software - system backup.

5. To provide the ability for single record updating and batching of updates.

6. Enable the user to query while updating.

Developed By B. White

CONSTRAINTS:

1. Rental should not exceed $15,000 per annum.
2. System employed should be maintained by the developer.
3. System should have been operational for two years in at least five industry-related data centers.
4. A technical user training program must be available.

Developed By B. White

ATE: 4/15/73

PAGE 1 OF 2

GOAL STATEMENT:

Developed By ____________

OBJECTIVES: (continued)

7. Convert existing programs to the new system with the aid (no cost) of the software firm.

8. Enable upper management to query a single record for specific information or, through the use of key words, receive an exception report.

Developed By B. White

CONSTRAINTS: (continued)

5. Must have a users group associated with the system.
6. A contract must be part of the seller's package.
7. System must run on our current equipment.

Developed By B. White

DATE: 4/15/73

PAGE 2 OF 2

FORM 5

LIST OF ASSOCIATED TASKS

pporting jective and/or onstraint Number	Evaluation Question Number	Evaluation Question	Required Answer	System X Answer	System Y Answer
1.Constraint	A1	What is the Direct Cost?	>15,000/yr	$25	$5,000
2.Constraint	A7h	The seller will provide maintenance	Yes	Yes	Yes
3.Constraint	B9	Total number of educational institutions using the system	5	100	48
4.Constraint	A7d	Will train user personnel	Yes	No	Yes
5.Constraint	B6	Is there a users group associated with the system	Yes	Yes	No
6.Constraint	F1	Is there a contract associated with the system	Yes	No	Yes
7.Constraint	C1	Equipment the system currently operates on	IBM/360 1130	IBM 1130	IBM 360
1. Objective	C5G	Are terminals available as input to the system	Yes	No	No
2. Objective & 3.	Accrued Total For Evaluation	Total must be more than 250 to qualify as a data management system	250	189	325
4. Objective	A5H	System maintenance will be provided	Yes	Yes	Yes
	D44	Statistical reports available	Yes	Yes	Yes
5. Objective	D34A	Can the system update single records	Yes	Yes	No
	D34D	Can the system hold and batch queries	Yes	No	No
Optional — List Accrued Totals For All Pages (FORM 5) If Weights Have Been Assigned To The Evaluation Questions				189	325

Note: Circle Any System's Answer Which is in Conflict With The Required Answer

DATE: 4/15/73 Developed By B. White PAGE 1 OF 2

LIST OF ASSOCIATED TASKS

pporting jective and/or straint Number	Evaluation Question Number	Evaluation Question	Required Answer	System NLHE-IS Answer	System SCORE-II Answer
6. Objective	D35	Can the system query while updating	Yes	No	No
7. Objective	A7E	Installation and conversion of existing systems	Yes	No	Yes
8. Objective	D24	Can data element retrieval take place	Yes	Yes	Yes
	D36A-J	Exception reporting possible	Yes	Yes	Yes
ptional - list Accrued Totals For All Pages (FORM 5) If Weights Have Been Assigned To The Evaluation Questions				189	235

Note: Circle Any System's Answer Which is in Conflict With The Required Answer

DATE: 4/15/73 Developed By B. White

PAGE 2 OF 2

FORM 4

	SYSTEM 1	WEIGHTS (OPTIONAL)	SYSTEM 2	WEIGH (OPTION
A. Cost 1. What is the direct cost (purchase price of this package?	purchase price	☐	purchase price	
2. Is the package available on a lease basis?	circle Yes or No	☐	circle Yes or No	
3. If the answer to question 2 is Yes, what is the lease price?	lease price per month $________	☐	lease price per month $________	
4. Are the options available (above) suitable for us to budget for at this time and into the future?	circle Yes or No	☐	circle Yes or No	
5. Is it possible to acquire this package in modules so that we need only acquire what we need at this time, yet add modules at a later date?	circle Yes or No	☐	circle Yes or No	
6. Can any of our lease costs be used toward possible subsequent purchase of the package?	circle Yes or No	☐	circle Yes or No	

PAGE TOTAL 1 ☐ PAGE TOTAL 2

FORM 5

	SYSTEM			SYSTEM	
	1	2		1	2
1			25		
2			26		
3			27		
4			28		
5			29		
6			30		
7			31		
8			32		
9			33		
10			34		
11			35		
12			36		
13			37		
14			38		
15			39		
16					
17			TOTAL FROM ALL PAGES		
18					
19					
20					
21					
22					
23					
24					

NOTE: TRANSFER THE ABOVE TOTAL(s) TO FORM 3

EXPERIENCES IN THE DEVELOPMENT OF MANAGEMENT INFORMATION SYSTEMS

Donald C. Bruegman
Director of Systems
University of Cincinnati

Experiences In The Development Of Management Information Systems

Donald C. Bruegman
University of Cincinnati

Introduction

Management information systems (MIS) are very popular today. They are much like the computer craze of the 1950's. Everybody thinks they should have one. Unfortunately, there is no model MIS against which success can be measured. The information needs and the management styles at each institution are so different that no one can really define a true MIS.

Many data processing professionals think that an MIS exists if reports are provided to management. This is an illusion. Reports must be used by management in the planning and strategic decision-making process before a true MIS can exist. If the reports are not being used in this manner, and are used only for solving day-to-day operational problems, the system is nothing more than a management reporting system. Too many institutions today have a management reporting system, and not a management information system.

The greatest problem in developing an MIS is the difficulty in determining what information management will need or use. Management cannot provide much help in this effort because they are constantly responding to different problems under changing sets of circumstances. This is what makes the MIS so elusive. The data processing professional is quickly frustrated when becoming involved in MIS development. He is too used to implementing day-to-day operational systems for middle and lower management where success is ordinarily guaranteed. In day-to-day operational systems the outputs are well defined in advance and the system must be used to get the day's work done even if it is poorly designed. Just the opposite is true of the MIS. The outputs are never well defined in advance and the system does not have to be used even if it is well designed.

There are two basic techniques for building MIS capabilities. One involves the top-down approach and the other is the bottom-up approach. Of course, neither is of any value unless the information is used by management. This is the risk in selecting either approach. The remainder of this paper will be devoted to a discussion of both of these approaches, the one selected at the University of Cincinnati, some examples of what has been accomplished using this approach, and finally, an evaluation of this decision.

About the University of Cincinnati

Before describing what has been accomplished it seems appropriate to provide some background about the institution. The University of Cincinnati is an urban university located in Cincinnati, Ohio. It is a state-assisted, public university with an enrollment of 36,000 students. There are sixteen colleges, including a medical college with a large teaching hospital, and several branch campuses. The total institutional budget is $130 million.

The data processing organization consists of the Systems Department which performs the systems development functions and the Southwestern Ohio Regional Computer Center (SWORCC) which provides computer services. SWORCC is a computer consortium organized and managed by the University of Cincinnati and Miami University at Oxford, Ohio (35 miles north of Cincinnati). SWORCC provides programming and computer services to the academic, research and administrative divisions of both institutions by means of an IBM 370/165. The system contains 3 million bytes of core, runs under OS/MVT with HASP and includes broad teleprocessing capabilities.

How to Create the MIS Data Base

To develop the MIS data base at the University of Cincinnati a bottom-up approach was selected. This involved using the data from existing day-to-day operational systems as the data base for the MIS. To accomplish this, it was necessary to implement a considerable number of basic operational systems.

The top-down approach involves determining MIS needs in advance and then building a data base to meet these needs. This approach is made more expedient today through use of a Data Base Management System (DBMS). Even prior to the existence of a DBMS the top-down approach was characterized by the need to anticipate in advance every conceivable use of data before undertaking the design and implementation of a system.

The bottom-up approach was selected at the University of Cincinnati for reasons which are perhaps more practical than theoretical. First, it was necessary to develop the MIS as soon as possible. The MIS requested was defined in general terms. Management desired information systems that would enable the university to better manage its resources. As was true at so many institutions at the time (and still is), sources of income were not keeping pace with inflation, plans for new programs and the commitments brought on by the expansion of the 1960's. Information was needed to evaluate competing requests for scarce resources.

The second reason the bottom-up approach was selected was because data were available from existing operational systems and these systems had been well integrated. Integration of one system with another had been planned when operational systems were first being developed in order to minimize redundancy of data among systems. (For example, an employee's social security number was carried in separate payroll, personnel and faculty effort reporting systems to identify a faculty member, but his name was carried only once in the payroll file.)

Urgency and the existence of a well integrated data base, therefore, are the reasons why the bottom-up approach was selected. Exhibit 1 provides a historical overview of how systems were developed at the University of Cincinnati using this approach. Note that the basic operational systems were developed starting in 1967 and the first MIS system was implemented in 1971.

Systems Design Guidelines

In utilizing the bottom-up approach to MIS development, certain systems design guidelines were followed when implementing operational systems and MIS systems. The following guidelines were followed when designing systems supporting day-to-day operations.

1. Concentrate on the basic systems first (e.g. accounting, payroll, personnel, student enrollment, budget, space, faculty effort)
2. Follow a logical implementation sequence (e.g. accounting and budget, personnel and faculty effort, student records and admissions)
3. Link systems and minimize redundancy of data
4. Use common codes
5. Provide "extract" capabilities to satisfy special reporting needs
6. Retain historical data
7. Think about MIS potential

When it came time to use the data from operational systems to create the data base for the MIS, another set of guidelines were followed when designing MIS systems. These included:

1. Build temporary data bases for large MIS applications
2. Provide data editing capabilities
3. Consider alternative data elements and/or logic to overcome data problems
4. Utilize task forces or committees to aid in the design of the MIS

5. Involve users responsible for operational data bases
6. Utilize summarization and responsibility reporting techniques
7. Acquaint everyone with quantification techniques

To demonstrate the bottom-up approach in the development of an MIS, two specific management information systems are described in the following sections. One is a cost study at the medical center and the other is an institutional program budgeting system.

Medical Center Cost Study

A Medical Center Cost Study was undertaken for several reasons. There was a desire by management to understand the cost relationships at the medical center, particularly between the Medical College and the teaching hospital. In addition, program costs were needed to support external funding arrangements.

The project was sponsored by the NIH and AAMC. The University of Cincinnati was one of thirty-five other schools performing the study. This provided an excellent opportunity to obtain comparative data from other institutions.

To give an idea of the type of information that was obtained from this study, a sample report is shown in Exhibit 2. This report provides costs per student, by unit of instruction, for each instructional program at a medical center.

The manner in which this system was developed is shown in Exhibit 3. Data from three operational systems -- accounting, personnel and payroll -- were used to create a cost study data base. This data base was used to drive the system.

Six months of elapsed time were required to complete the systems and programming involved. The project itself spanned about one and one-half years from July 1971 to November 1972. The manpower involved included thirty weeks of systems and twenty weeks of programming. The implementation cost was $82,000. This cost includes systems, programming, computer services, data collection and all user time associated with the project.

Program Budgeting

Another MIS project which has been undertaken is program budgeting. This system is being developed in order to integrate planning with budgeting and thus improve the allocation of resources when preparing the institutional budget.

The Program Budgeting system is composed of two major reporting systems. One converts the current year's appropriation budget to a program budget and produces a Program Budget Submittal Form. See Exhibit 4. The other reporting system is a Program Profile. This system provides data concerning current program costs. See Exhibit 5. The Program Profile report is supported by detailed listings of faculty and courses.

The logic for developing the Program Budgeting system is shown in Exhibit 6. This system is much more complex than the Medical Center Cost Study, but the same principle is followed. Data are selected from operational systems to create the program budgeting data base. In this case, several intermediate data bases are created and brought together to produce the desired reports.

Although the program budgeting system is not formally implemented, it was pilot tested this past summer. This pilot test involved converting last year's appropriation budget to a program budget and distributing the Submittal Forms and Program Profiles to the colleges and departments so that they could review the data and comment on the system. The system will be used this year to prepare the 1975-77 biennial budget.

The Program Budgeting System required one and one-half years to design and implement. The project started in December 1971, and the material was distributed for pilot testing in May 1973. Six months of elapsed time were required to complete the computer system. 142 weeks of systems and 104 weeks of programming manpower were required. The implementation cost of the project was $209,000.

Evaluation

Looking back over the seven year period from fiscal year 1967 through 1973, the estimated costs and benefits of the accomplishments during this period pertaining to both operational systems development and MIS activities can be compared as follows:

Costs:

Systems	$1,100,000
Data Processing	6,500,000
	$7,600,000

Benefits:

Tangible:	Increased Income	$4,000,000
	Direct Cost Savings	$1,000,000
Intangible:	Satisfy administrative requirements	?
	Improved service	?
	Increased efficiency	?
	Value of information	?

The dollars assigned to the intangible benefits determine the degree of success achieved. Inasmuch as these intangible benefits are the factors which justify most systems, they deserve further explanation.

To "satisfy administrative requirements" means that one has met the payroll, conformed to state and federal reporting requirements, registered students, etc. The list can go on and on. At a large institution much of this work can only be accomplished through use of a computer.

"Improved service" involves going one step beyond satisfying the administrative requirements described above. It includes, among other things, getting the payroll out on time, producing accounting reports early in the month, and eliminating registration lines.

"Increased efficiency" relates to such matters as reduced workloads, clerical savings, streamlined procedures and minimization of red tape and bureaucracy.

The "value of information" relates directly to the MIS and is perhaps the most difficult benefit to assign a dollar value. If management decisions are based on the information provided, and the decisions are good ones, there is a high value; otherwise there is little or no value. At the University of Cincinnati several million dollars have been cut from the budget over the past few years without any major program reductions. What role the MIS has played in this can be debated, but the information provided by the MIS has been used during budget deliberations.

To evaluate the bottom-up approach used to build the MIS at the University of Cincinnati the advantages and disadvantages of this approach are described below.

The advantages would include:

1. Supports day-to-day operations. Operational priorities are satisfied. There are minimal complaints from middle and lower management that resources are being diverted from day-to-day operations which keep the institution going.
2. Single data base. When there is a separate data base for operations and another for MIS, problems of data duplication and data accuracy occur. Even with conversion to a Data Base Management System (DBMS) there may still be two separate data bases, one for the systems converted to the DBMS and another for the systems not yet converted.

3. Less costly. By concentrating on operational systems there is always a potential cost savings which can be achieved. An independent MIS or conversion of existing operational systems to a Data Base Management System does not guarantee such a return. Also, by using the existing operational data base it is possible to test management's use of information without making a substantial investment in MIS development.
4. Enhances creditability. By developing operational systems first, and then moving toward MIS, one can demonstrate his capabilities, instill confidence and rally support for the development of advanced information systems.
5. Greater MIS potential. The number and types of operational systems which have been implemented have a direct bearing on the scope and flexibility of the MIS. As this number increases, the potential of the MIS data base expands. Also, by utilizing the operational data base there is a faster start-up when developing the MIS.

Among the disadvantages of the bottom-up approach are:

1. Problems with accuracy of data and missing data elements. This factor is an ongoing problem. When operational systems are designed, emphasis is always on day-to-day needs and day-to-day expediencies. Not only are some data overlooked, but some data are intentionally excluded.
2. Limited management involvement. Although top management involvement is difficult to obtain in either approach, the bottom-up approach limits the MIS data base to the data available from operational systems. This creates a tendency to compromise some of top management needs when designing the MIS.
3. Responsibility for MIS not fixed. By concentrating on operational systems the organizational structure is "operations" oriented. As a result, there is no one to manage the MIS.

4. Potential political problems. By having an abundance of data available from operational systems, and forging ahead with limited management involvement, it is possible to provide too much information too fast. This frightens people and sometimes even makes them anti-MIS.

5. Limits use of data base management software. When operational systems already have been implemented and the MIS is built upon this data base, the conversion to a Data Base Management System (DBMS) is questionable and costly. As a result one is not afforded the technical advantages of utilizing a DBMS.

Up to now it has been concluded that the advantages of the bottom-up approach outweigh the disadvantages. At the University of Cincinnati this position is constantly being evaluated. If the University of Cincinnati can benefit from a change in approach, that change will be made. This is really another way of saying that MIS development is an institutional decision. In developing the MIS, one must do everything possible to make sure that a management information system, and not a management reporting system, is being developed. The approach that is selected depends upon the level of sophistication of the institutional data base and the urgency of the task. The top-down approach is perhaps best when just starting out or there is no sense of urgency. Otherwise, the bottom-up approach offers a great deal of potential.

EXHIBIT 1

UNIVERSITY OF CINCINNATI
HISTORY OF SYSTEMS DEVELOPMENT

	MAJOR OPERATIONAL SYSTEMS	MIS SYSTEMS
1967	1. Accounting	
1968	2. Payroll	
	3. Student Records & Registration	
	4. Student Aid Awards	
	5. Budget	
	6. Space	
1969	7. Personnel	
	8. Alumni	
	9. Admissions	
	10. Hospital: SHAS	
1970	11. Faculty Effort Reporting	
	12. Housing	
	13. Development	
	14. Hospital: Ancillary Services	
1971	15. Grants & Contracts	1. Departmental Profiles
	16. Improved Registration System	
	17. Hospital: Personnel Reporting System	
1972	18. Courses Data Base	2. Medical Center Cost Study
		3. Profitability Statements
		4. Monitor Appropriations
1973	19. Student Aid Applications	5. Program Budgeting
	20. New Student Records & Registration	6. Financial Management Reporting System
	21. Student Billing	
	22. Affirmative Action	

SAMPLE DATA

MEDICAL CENTER PROGRAM COSTS PER STUDENT

INSTRUCTIONAL PROGRAMS	NUMBER OF STUDENTS	<-----------COSTS PER STUDENT BY UNIT---------->					MEDICAL CENTER TOTAL COSTS	OUTSIDE CONTRIBUTIONS		TOTAL COSTS PER STUDENT
		COLLEGE OF MEDICINE	COLLEGE OF NURSING & HEALTH	COLLEGE OF PHARMACY	TEACHING HOSPITAL	PRIVATE HOSPITAL		AFFIL. UNITS	MEMO SUMMARY	
UNDERGRAD MEDICAL	427	$ 3,375	$ 0	$ 0	$ 1,278	$ 0	$ 4,656	$ 511	$ 709	$ 5,876
UNDERGRAD NURSING	383	$ 122	$ 1,763	$ 0	$ 387	$ 0	$ 2,275	$ 306	$ 922	$ 3,503
UNDERGRAD PHARMACY	289	$ 19	$ 29	$ 1,364	$ 108	$ 13	$ 1,534	$ 62	$ 814	$ 2,410
GRADUATE MEDICAL	109	$ 12,932	$ 0	$ 0	$ 39	$ 0	$ 12,973	$ 603	$ 68	$ 13,644
GRADUATE NURSING	28	$ 45	$ 7,697	$ 0	$ 420	$ 0	$ 8,165	$ 0	$ 1,786	$ 9,951
GRADUATE PHARMACY	20	$ 124	$ 0	$ 2,352	$ 1,441	$ 0	$ 3,919	$ 0	$ 941	$ 4,860
OTHER ALLIED HEALTH	15	$ 10,673	$ 0	$ 3	$ 18,024	$ 0	$ 28,702	$ 315	$ 0	$ 29,017
OTHER DEGREE CREDIT	26	$ 5,064	$ 0	$ 0	$ 5,051	$ 0	$ 10,118	$ 917	$ 12	$ 11,047
INTERNS & RESIDENTS	399	$ 4,952	$ 5	$ 0	$ 2,146	$ 19	$ 7,124	$ 2,833	$ 625	$ 10,582
POST DOCTORAL MEDICAL	96	$ 5,078	$ 0	$ 0	$ 33	$ 0	$ 5,113	$ 44	$ 293	$ 5,450
CONTINUING ED MEDICAL	2,152	$ 116	$ 0	$ 0	$ 9	$ 0	$ 127	$ 8	$ 1	$ 136
CONTINUING ED NURSING	1,127	$ 2	$ 68	$ 0	$ 5	$ 0	$ 77	$ 0	$ 0	$ 77
CONTINUING ED PHARMACY	180	$ 0	$ 0	$ 101	$ 0	$ 0	$ 103	$ 58	$ 0	$ 161
OTHER NONDEGREE CREDIT	11	$ 4,727	$ 0	$ 73	$ 17,606	$ 0	$ 22,408	$ 429	$ 0	$ 22,837

EXHIBIT 3

MEDICAL CENTER COST STUDY LOGIC

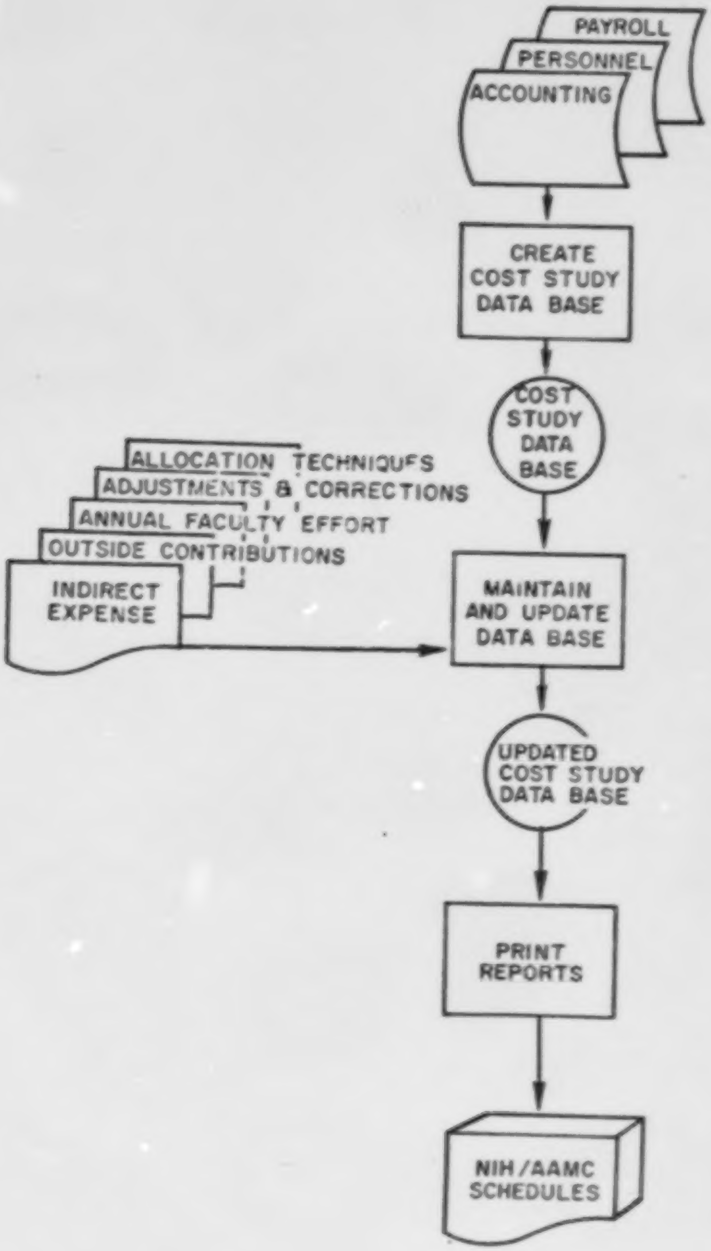

PAYROLL
PERSONNEL
ACCOUNTING
CREATE COST STUDY DATA BASE
COST STUDY DATA BASE
ALLOCATION TECHNIQUES
ADJUSTMENTS & CORRECTIONS
ANNUAL FACULTY EFFORT
OUTSIDE CONTRIBUTIONS
INDIRECT EXPENSE
MAINTAIN AND UPDATE DATA BASE
UPDATED COST STUDY DATA BASE
PRINT REPORTS
NIH/AAMC SCHEDULES

EXHIBIT 4

DEPARTMENT : SOIL CONSERVATION
DIVISION : AGRICULTURE
(002001)

UNIVERSITY OF CINCINNATI
PROGRAM BUDGET SUBMITTAL FORM

DATE : 4-15-73
PAGE : 1

PROGRAM			BUDGETED AMOUNTS					TOTAL	FUND SOURCES				
												GRANTS AND CONTRACTS	
CODE NAME	STUDENT FTE	FACULTY FTE	ASSIGNED FACULTY COMPENSATION	UNASSIGNED FACULTY COMPENSATION	OTHER COMPENSATION	NON RECURRING EQUIPMENT	OTHER EXPENDITURES		GENERAL	DESIGNATED	RESTRICTED	AWARDED	APPLIED FOR
BAC PROF 1972-73	60.0	3.0	51,000	5,355	15,000	4,500	12,000	87,855					
1973-74	40.0												
1974-75	50.0												
MASTERS 1972-73	15.0	1.5	30,750	2,678	7,500	2,250	6,000	49,178					
1973-74	20.0												
1974-75	30.0												
SPONS RES 1972-73		2.0	42,000	3,570	10,000	3,000	8,000	66,570					
1973-74													
1974-75													
PUBLIC SVC 1972-73		.5	10,000	892	2,500	750	2,000	16,142					
1973-74													
1974-75													
STDNT SUPT 1972-73		.5	8,900	892	2,500	750	2,000	15,042					
1973-74													
1974-75													
DEPT ADMIN 1972-73		1.0	23,000	1,785	5,000	1,500	4,000	35,285					
1973-74													
1974-75													
TOTAL 1972-73	75.0	8.5	165,650	15,172	42,500	12,750	34,000	270,072	235,072		10,000	25,000	
1973-74	60.0												
1974-75	80.0												

DEPARTMENT : SOIL CONSERVATION
COLLEGE : AGRICULTURE

PROGRAM PROFILE

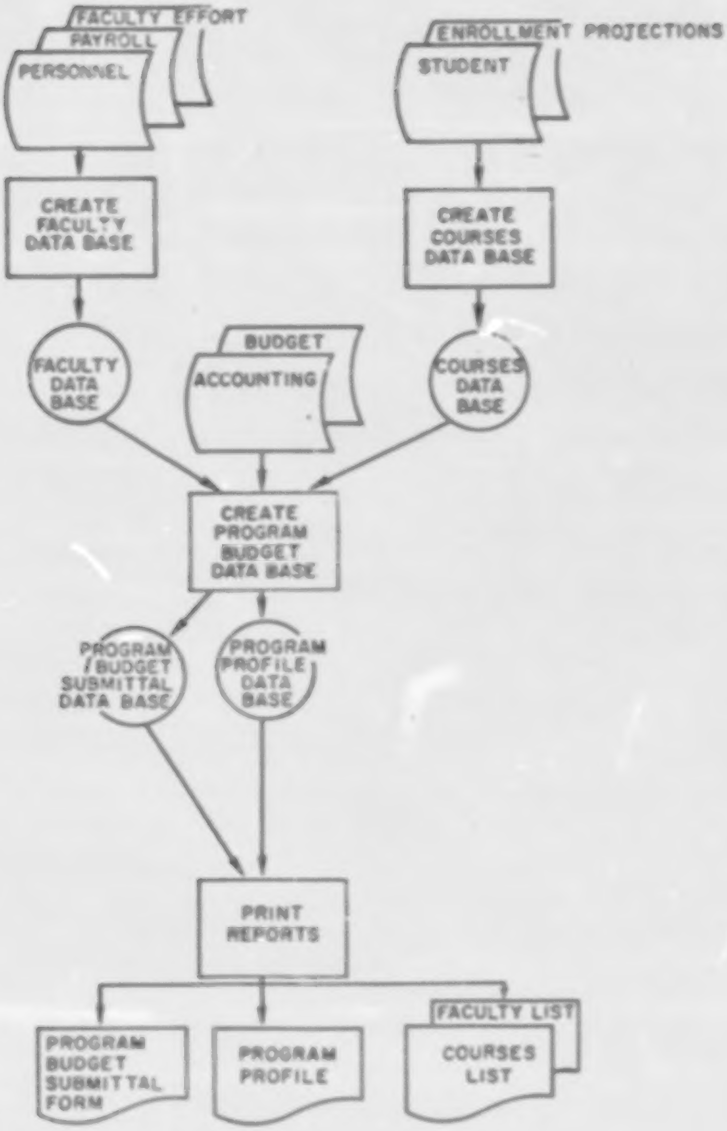
AUTUMN QUARTER 1972

DATE : 4-15-73
PAGE : 1

PROGRAM	STUDENT CR HRS	STUDENT FTE	FAC FTE	STU/FAC FTE RATIO	BUDGETED EXPEND PER STUDENT FTE	% OF TOTAL DEPT BUDGET	AVG ANNUAL FACULTY COMP	AVG FACULTY EFFORT %	REPORTED AVG WEEKLY FACULTY FTE LOAD WK CR HR	WK CN HR	IND STU HRS	STD CRD HRS
BACC PACE ACTUAL 1972-73	900	60.0	3.0	20.0:1	1,464	32.4	17,000	35.3	12	14	1	264
MASTERS ACTUAL 1972-73	225	15.0	1.5	10.0:1	3,279	18.2	20,500	17.6	10	11	3	227
SPONS RES ACTUAL 1972-73			2.0			24.7	21,000	23.5				
PUBLIC SVC ACTUAL 1972-73			0.5		.	6.0	20,000	5.9				
STDNT SUPT ACTUAL 1972-73			0.5			5.6	17,800	5.9				
DEPT ADMIN ACTUAL 1972-73			1.0			13.1	23,000	11.8				
DEPARTMENT TOTAL ACTUAL 1972-73	1,125	75.0	8.5		4,333	100.0	19,488	100.0	6	7	1	133
ACTUAL 1972-73 DEPARTMENT VALUES FOR COMPARISON TO CBR MODEL				11.5:1	2,713		19,023					133
CBR MODEL VALUES FOR 1972-73				11.5:1	1,888		16,308					173

EXHIBIT 5

PROGRAM BUDGETING LOGIC


FACULTY EFFORT
PAYROLL
PERSONNEL
ENROLLMENT PROJECTIONS
STUDENT
CREATE FACULTY DATA BASE
CREATE COURSES DATA BASE
FACULTY DATA BASE
BUDGET
ACCOUNTING
COURSES DATA BASE
CREATE PROGRAM BUDGET DATA BASE
PROGRAM BUDGET SUBMITTAL DATA BASE
PROGRAM PROFILE DATA BASE
PRINT REPORTS
PROGRAM BUDGET SUBMITTAL FORM
PROGRAM PROFILE
FACULTY LIST
COURSES LIST

VIRGINIA TECH ADMINISTRATIVE MANAGEMENT INFORMATION SYSTEM

A. Wayne Donald
Project Manager
Computing Center
Virginia Polytechnic Institute
and State University

TABLE OF CONTENTS

I. INTRODUCTION 4

  A. Virginia Polytechnic Institute and State University 5
  B. University Administrative Organization
  C. Computing Center 6

II. OFFICE OF SYSTEMS DEVELOPMENT 8

  A. History and Objectives 8
  B. Organizational Chart 9

III. DATA BASE MANAGEMENT 10

  A. The Need for Data Management 10
  B. Selection of IMS 11
  C. Data Base Administrator 12

IV. SYSTEMS DESIGN AT VIRGINIA TECH 16

  A. Development Cycle 16

V. UNIVERSITY APPLICATIONS 19

  A. Accounting 19
  B. Library 20
  C. Student Records 22

    1. Admissions Subsystem 22
    2. Registration Subsystem 23
    3. Student Accounting Subsystem 24

  D. Personnel/Payroll 25

    1. Personnel 26
    2. Payroll 26

  E. Other Productive Systems 27

    1. Research Proposal Monitor System 27
    2. Gift Giving System 28
    3. Alumni System 29
    4. Space Utilization 30

IV. FUTURE DEVELOPMENTS AT VIRGINIA TECH 31

A. Facilities 31
B. University Fixed Assets 32
C. University Book Store 32
D. Purchasing 33
E. Test Creation and Evaluation 34
F. Graduate School System 34
G. Other Systems for Consideration 34

VII. APPLICATION INTEGRATION 36

A. Present Position 36
B. Integration Plan 38
C. Accomplishment of Application Integration 39
D. Future Plan 41

INTRODUCTION

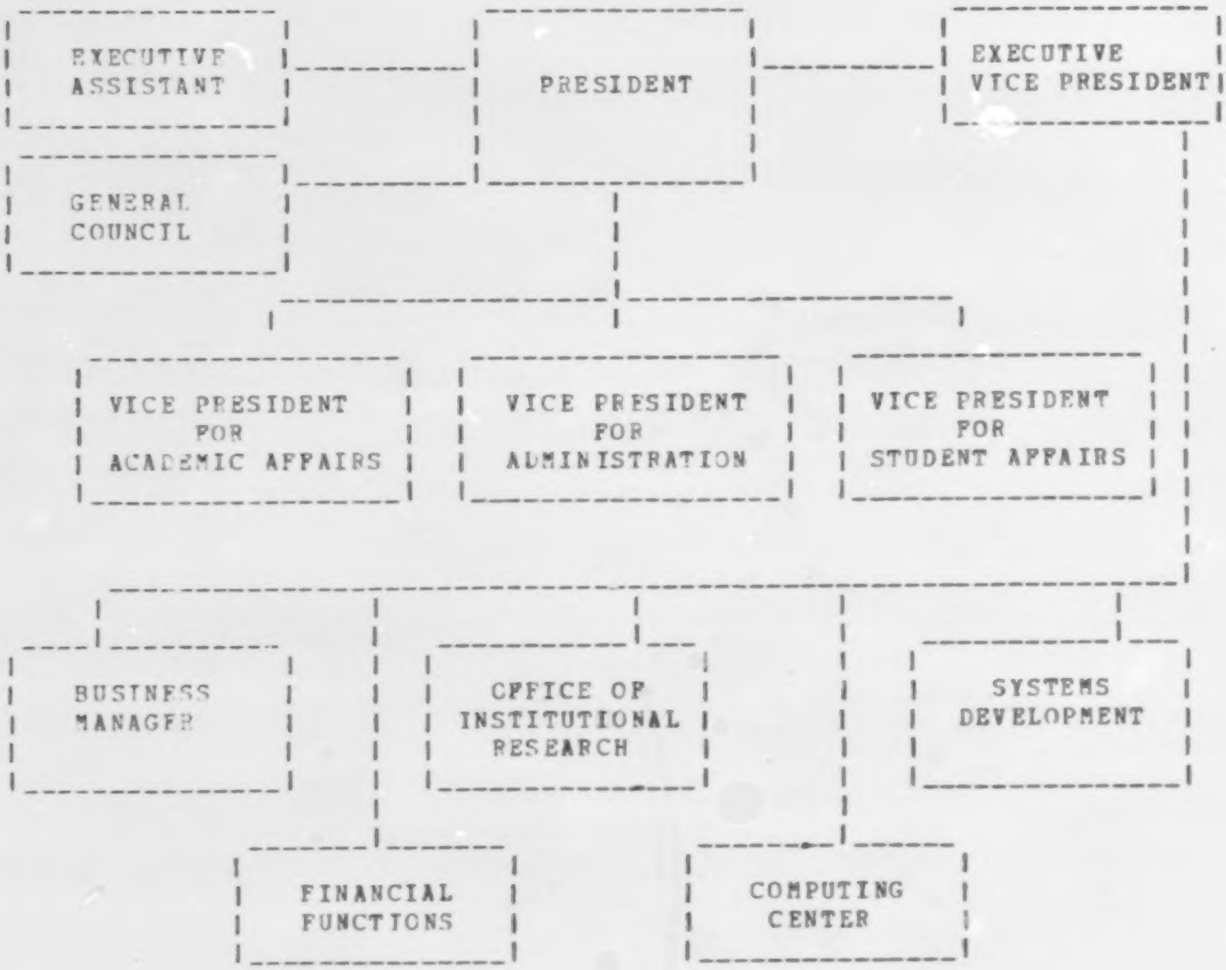
Virginia Polytechnic Institute and State University

Virginia Polytechnic Institute and State University was established as Virginia's land-grant college in 1872. It is located in scenic southwest Virginia, adjoining the town of Blacksburg. The campus lies on a plateau between the Blue Ridge and Allegheny Mountains, 2,100 feet above sea level. The area is noted for its natural beauty, healthy atmosphere, and outdoor recreational opportunities.

Virginia Tech's student body is coeducational, numbering about 5,000 women students among its 16,000 total.

Today, Virginia Tech serves the state, the nation, and the world in the three important fields of instruction, research, and extension.

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY ADMINISTRATIVE ORGANIZATION


EXECUTIVE ASSISTANT
PRESIDENT
EXECUTIVE VICE PRESIDENT
GENERAL COUNCIL
VICE PRESIDENT FOR ACADEMIC AFFAIRS
VICE PRESIDENT FOR ADMINISTRATION
VICE PRESIDENT FOR STUDENT AFFAIRS
BUSINESS MANAGER
OFFICE OF INSTITUTIONAL RESEARCH
SYSTEMS DEVELOPMENT
FINANCIAL FUNCTIONS
COMPUTING CENTER

VIRGINIA POLYTECHNIC INSTITUTE AND
STATE UNIVERSITY
COMPUTING CENTER
ORGANIZATIONAL CHART

```
                              ------------
                             |            |
                             | DIRECTOR   |
                             |            |
                             |____________|
                                   |
                                   |
                              ------------
                             |            |
                             | ASSOCIATE  |
                             | DIRECTOR   |
                             |____________|
                                   |
      -----------------------------|-------------------------
      |                |                  |                  |
   ---|----------   ---|-----------   ----|---------    -----|----------
  |              | |               | |               | |                |
  |   MANAGER    | |    MANAGER    | |    MANAGER    | |    MANAGER     |
  |   EDUCATION  | | PUBLICATIONS  | |  CONSULTING   | |    COMPUTER    |
  |              | |               | |               | |    SERVICES    |
  |______________| |_______________| |_______________| |________________|
```

The Virginia Tech Computing Center supplies computing facilities and computing information to the University's academic, research, administrative, and extension users.

The Computing Center has two IBM System/370 Model 158 processors connected by a channel-to-channel adapter which permits direct communication between the two otherwise independent processors. The main processor runs under control of Operating System/370 Multiprogramming with a variable number of tasks (OS/MVT), and performs job execution;

the support processor runs under IBM's Attached Support Processor (ASP) and OS/MVT and performs input/output, scheduling functions, and some job execution.

HARDWARE AVAILABLE

MODEL	DESCRIPTION	NO.
370	Model 158 (1.5 and 2.0 megabytes)	2
3420-3	9-Track Tape Unit	5
2501	Card Reader	1
2540	Card Read/Punch	1
3330	Disk Drive	16
1627-1	Plotter (12")	1
1627-2	Plotter (30")	1
1403-N1	Printer	1
3211	Printer	1
2495	Tape Cartridge Reader	1
3270	Video Display Terminals	19
3284	Printing Terminal	2
2741	Communications Terminal	5
DATEL	Communications Terminal	13

Additionally, various terminals are installed at community colleges and other educational institutions in the state.

OFFICE OF SYSTEMS DEVELOPMENT

History and Objectives

In the University's efforts to provide adequate computational facilities and services, to develop computer-based information systems, and to contribute to the statewide development of computer networks, the Systems Development Division was shifted from the Computing Center to a staff office reporting to the Executive Vice President. The office remains a member of the Executive Committee of the Computing Center.

The Office of Systems Development is responsible for analyzing present and future data processing requirements for the University. A request for a system is initiated by this staff office, beginning a development cycle. In addition to the Director of Systems Development, there are sixteen other staff members. Figure 2.1 shows an organizational chart.

A request is assigned to the appropriate Project Manager, who works with the user office and starts the development cycle. The Project Manager is responsible for the complete systems design.

VIRGINIA POLYTECHNIC INSTITUTE AND
STATE UNIVERSITY
OFFICE OF SYSTEMS DEVELOPMENT
ORGANIZATIONAL CHART

```
                                         -------------
                                        |             |
                                        |  DIRECTOR   |
                                        |             |
                                        |_____________|
                                               |
                                               |
                                               |
                                               |
               --------------------------------|--------------------------
              |                |               |               |          |
        ------|-------   ------|-------   -----|-------   -----|----   ----|-----
       |    DATA      | |   ANALYST    | |   PROJECT    | |PROGRAMMER| | ANALYTIC |
       |    BASE      | |    POOL      | |   MANAGERS   | |   POOL   | | STUDIES  |
       |ADMINISTRATOR | |              | |              | |          | |          |
       |______________| |______________| |______________| |__________| |__________|
                                               |
                                               |
                                               |
                                               |
               --------------------------------|--------------------------
              |                |               |               |          |
        ------|-------   ------|-------   -----|-------   -----|----   ----|-----
       |  FACILITIES  | |  FINANCIAL   | |   STUDENT    | |PERSONNEL | | LIBRARY  |
       |              | |              | |   RECORD     | | PAYROLL  | |    .     |
       |______________| |______________| |______________| |__________| |__________|
                                               |
                               ----------------|--------------
                              |                               |
                        ------|-------   --------------   ----|---------
                       |   STUDENT    | |  ADMISSIONS  | |REGISTRATION|
                       |  ACCOUNTING  | |              | |            |
                       |   ___________| |______________| |____________|
```

Figure 2.1

DATA BASE MANAGEMENT

The Need for Data Management

Traditionally, as the need for computerization within colleges and universities became evident, each application area was automated independently of existing or planned application areas. Eventually, many colleges and universities possessed files for the various applications containing much information which was duplicated in other files. In an attempt to satisfy the requirements of a new system without defining a new file containing redundant information, existing files were expanded. A new problem arose: the programs which accessed the old files had to be modified to accommodate the expanded formats. The solution seemed to lie in the definition of a supervisory data management system, under which all data could be defined. The new system would: eliminate redundant information; provide a facility for data base expansion (transparent to the users of the original data base); and allow rapid access to specific data elements, both in an off-line batch mode and in a telecommunications mode through many terminal types.

The Office of Systems Development at Virginia Tech recognized the need for a data management facility early in the development of a university-wide information system. The only existing computerized applications used first or second generation concepts and very little of these systems were salvaged. Therefore, plans were made to develop all administrative applications under control of a data management system.

Selection of IMS

Many data management packages were investigated. It was discovered that none of the available systems provided all of the desirable facilities. IBM's Information Management System (IMS) incorporated the most important features. Specifically, the system provides a means of establishing data bases in terms of hierarchical tree structures to isolate segments of related information. These segments rely on a higher level segment for complete definition. Redundant information is thereby reduced. Each segment of information is considered to be made up of fields. The highest level segment is the root segment. Dependent segments rely on some higher level segments for full meaning or identification. This structure provides an effective method of handling variable-length data by allowing non-occurrence or multiple occurences of segment types. Separate data bases may be interrelated through logical relationships, allowing each user to define data bases composed of any segments of his choosing regardless of the actual physical structures. Thus, data bases may be restructured and expanded without modification to the existing application programs which use the data bases.

Data base organization can occur several ways and depends on the data access method. The application programs are independent of the physical organization of the data bases. Sequential, indexed or direct techniques may be used to access data. IMS incorporates a telecommunications capability which provides access to data from remote terminals as well as access in a batch environment. The telecommunications interface is provided in the IMS control software so that application programs which access the terminals may be written in a high-level language such as COBOL or PL/1. Necessarily, IMS also provides security, check-point, restart, data bases recovery and reorganization, and statistical capabilities for continued maintenance and analysis of the systems environment.

Data Base Administrator

The individual Project Manager is responsible for the complete design of his system. This includes standards, data base structure, integrity, storage environment, and the operating environment. With the increase in the number of applications at Virginia Tech, some central control needed to be established to assist the Project Manager in these areas of responsibility.

The first step was the creation of a data base committee to review all data bases in production as well as those in the design phase. The committee assisted the Project Manager in the design of his data bases so that an efficient structure, benifiting the user, could be built. The committee also reviewed the data elements in each data base and found that many elements existed with the same meaning but different names. It was extremely important to establish some naming standards for data elements.

After reviewing the existing problems, the Director of Systems Development created a position for a Data Base Administrator. The position has responsibilities in three general areas. The first is: the content, organization, integrity and privacy of the data base. The physical data storage environment and operating environment of the data base, (including user interface, monitoring, scheduling, and recovery) are two more areas of responsibility. Within each of these three areas, specific functions of the data administrator can be itemized.

Content, Organization, Integrity, Privacy

1. Defines the content and organization of the data base by assisting application developers in segment structure, selection of access methods, and definition of logical relationships.
2. Documents the data base by providing such tools as a data element dictionary and automated data base layout facility.
3. Controls access to the data base by establishing security procedures.
4. Defines standards for data element names and definitions.
5. Insures the integrity of the data base by defining adequate backup and recovery procedures.

Physical Data Storage Environment

1. Defines the strategy for allocation of physical storages.
2. Develops procedures for determination of future storage requirements.

Operating Environment

1. Assists in evaluation and selection of software and hardware in support of the data base.
2. Works with application developers to consult with users in determining methods of data organization, retrieval, and extraction.
3. Monitors data base usage in segment access evaluation and response statistics.
4. Provides education for both application developers and users in data base concepts.
5. Disseminates information relating to procedures, techniques and suggestions for data base use.
6. Defines operational procedures which affect the data base.
7. Establishes a valid and secure environment for testing new application programs.
8. Defines the interface through which the user communicates with the data base by providing tools such as a general inquiry facility.

With the large number of responsibilities assigned to the Data Base Adminstrator, several committees were established for assistance in the TMS environment. A data base committee, with a multi-level structure, has as its charge the consideration of all matters concerning the total University Data Base. In recognition of future problems, a group of people was selected to consider questions of data security and privacy, resolutions of data access questions, and data creation and dissemination responsibilities. The group includes representatives from vice presidential offices, colleges, and the student body. The group meets to address problems as the need arises.

A data base design committee, consisting of all the Project Managers and the Programming Supervisor, was formed to assist in the design of all data bases. The members meet to discuss design structure and to help select the most efficient access method. Since the members of the committee are familiar with all data bases in production and those being designed, the meetings have reduced the amount of redundant information, and assisted in establishing logical relationships.

A temporary committee was created to address the immediate needs of establishing data element naming conventions, standardization of data element definitions, and standard abbreviations. Standards were also specified for the use of terminals, such as format naming conventions, use of functional keys, and standard screen formats. Since COBOL is the primary language used at Virginia Tech for accessing data elements, the COBOL restrictions were imposed for the standards. In addition to the above, the committee imposed some COBOL coding standards, especially for IMS programs.

A data base sub-committee was established with members representing the various Computing Center Divisions, as well as the Office of Systems Development. The function of this group is to resolve technical questions in the IMS area. They meet on a regular schedule to discuss new developments and any current problems. The results of the meeting are available for distribution to IMS users.

These various committee chairmen submit reports of each committee meeting to the Data Base Administrator. The Data Base Administrator and each committee chairman discuss each report.

SYSTEMS DESIGN AT VIRGINIA TECH

Development Cycle

The development cycle for an application begins when the Office of Systems Development receives a request for a proposal from a University office. Figure 4.1 indicates the various stages in the development cycle.

The Director assigns the user request to the Project Manager responsible for that particular area. The Project Manager will work with the user office to prepare a general proposal on how the request can be fulfilled. This proposal attempts to outline in broad terms the time and manpower requirements necessary to produce the system. Cost estimates are included for both the development of the system as well as for it's operation once it is turned over to the user office. In most cases, the data base(s) for the system is designed, both structure and contents, and included as an appendix to the proposal. This is helpful when discussing the system with the user office. They can see what information will be available and offer suggestions for modifications. The time period for this scope-of-effort study can vary, but usually requires from two to eight weeks. After the study is completed, the written proposal is presented to the user office for consideration.

If the user approves the proposal and wishes to proceed, the user office will provide Systems Development with the sufficient funds. The user office will designate his system coordinator to participate on the project team, and the detailed design of the system begins.

```
USER                       SYSTEM                     SYSTEM                    SYSTEM
REQUEST                    REVIEW                     DETAIL                    TURNOVER
  |                        PROPOSAL                   PROPOSAL                     |
  |                           |                          |                         |
  |                           |                          |                         |
  |                           |                          |                         |
  |                           |                          |                         |
  |    SCOPE-OF-              |    MAJOR                 |     PROGRAMMING         |
  V    EFFORT STUDY           V    DEVELOPMENT           V    ----------------     V
                                                           |______________________
 -----------------------------------------------------------| DOCUMENTATION
                                                           |________________
                                                           | EDUCATION
                                                           |________________
```

Development Cycle
Figure 4.1

This phase of the development includes data element dictionary specifications, program module outlines, anticipated data collection techniques, system output functions, and selection of any required hardware or software. At the end of this phase, a joint Systems Development - User Office proposal is presented for approval. The proposal is also referred to higher administration for approval.

When approval is received, the project team initiates an intense period of development. Program specifications are written and turned over to the supervisor, who has the responsibility of assigning the programs. The Project Manager has to coordinate many activities during this phase of the development cycle. Documentation, education, and complete test plans have to be initiated and carried on during this period. The system coordinator from the user office checks all the results during the test period and is responsible for their approval. After the test period has been successfully completed and the documentation approved, the Project Manager prepares to turn the system over to the user office. This turnover procedure attempts to establish whether or not the original design specifications were successfully met. Once the specifications are met, the system is delivered to the user office for operation.

In order for such a development cycle to be successful, there has to be a strong joint effort between Systems Development and the user office. A user is receiving a system that he asked for since he was actively involved in the design and development.

UNIVERSITY APPLICATIONS

Accounting

The Virginia Tech Accounting System was the first application developed under IMS. It consisted of two data bases and recorded the majority of University expenditures and revenues. The initial proposal called for the system to be developed in three phases. With improvements in hardware and software technology, Systems Development decided not to proceed with the final phase. Instead, the accounting system was reviewed, and it was subsequently decided to design a new system that would completely automate the Accounting Office, using as much of the existing procedures as possible.

The new system, which is scheduled for completion in September of 1974, will consist of seven data bases, which are:

Appropriation (Expenditures)
Revenue
Accounts Receivable
Local Funds
General Ledger
Appropriation History
Revenue History

The inputs to these data bases will retain the same format as in the existing system. There will be additional transaction codes for some of the new areas.

Terminal usage will increase considerably in the new system with terminals present in all colleges. Inquiry for any financial records associated with the college will be permitted by Deans. Terminal input will increase in the Accounting Office with data base maintenance, budget allotments, and transaction error corrections being the main areas utilized. The system has been designed to allow for future transaction entry via the terminal.

The hard copy outputs of the system will have a few changes. With the addition of five data bases, there will be several new reports that will be useful to the University administration. Many of the internal reports will be in the form of Computer Output Microfilm, which is presently used.

The accounting system maintains a financial record for every department and project on campus. With the additional data bases, more financial data will be available, which will allow for greater interfaces with other systems.

Library

A proposal was presented to the University Library in August 1970, to improve the level of service offered to the library user as well as to facilitate internal technical processing. The goal was to design a real-time library information system with two purposes: to provide sufficient timely information to influence the daily decision-making process; and to provide full interface capability to a University-wide information system.

At the present time, the acquisition portion of the system is productive. It has been turned over to the library personnel, who have the responsibility for its maintenance. The remaining portions of the system will be written and implemented at their request.

Order recommendations are received from the University departments and checked for duplication against information already in the system and checked for sufficient funds in the account of the requestor. Listings are provided to indicate refusals and possible duplicates. If the order is accepted, the appropriate accounting record is updated and a purchase order is produced.

A terminal is used for inquiry, entry of data pertaining to books on order and books received, and entry of accounting information.

When a book arrives in the library, it is first checked to see whether it was originally placed on order rather than being part of a blanket order or a gift. Once a determination has been made, the proper data bases are updated and the department can be informed of the arrival of the order. Entry of gifts and blanket orders are entered via the terminal in very much the same manner.

Student Records

The Student Records System at Virginia Tech was the largest system proposed by Systems Development. Its goal was to completely automate the record keeping involved for each student at the University. The system was so complex that it was split into three main areas, with one person responsible for each area. The following few pages will briefly define each of these three areas.

Admissions Subsystem

The basic objective of this system was to have the computer perform as much of the routine work as possible. The system is operative for both the Undergraduate and Graduate Admission Offices. Whenever an application is received it is entered into the system via a terminal. The application will be checked against an existing Scholastic Achievement Test data base, and any data will be displayed for verification. After all the required data has been entered, the record will be added to the data base.

The system can provide the Admissions Office with an "Applicants for Review" report which is used to determine acceptance or denial of admissions. The decision is entered through the terminal and the record updated. This will trigger several reports from the system, such as "Notice of Acceptance of Admission," "Housing Application Contract Card," "Response to Offer of Admission Form," and several others to be used in the admission process.

After a form has been received from a prospective student, the record is updated and data is available for other portions of the Student Records System. After the data base has had the final updates, it is available for several statistical reports for use by University and state officials.

Registration Subsystem

This portion of the system will begin with the preparation of the "Timetable of Classes" for each quarter. Academic departments will specify the required information about each section of each course on Timetable OpScan sheets. These will be forwarded to the Registrar's Office for review, and will then be processed to create the Timetable data base. After conflicts have been resolved, the Timetable will be printed for distribution.

Students will use the Timetable and a pre-printed OpScan form for pre-registration. After working out a class schedule with their advisor, the data will be put on the OpScan form and returned to the Registrar's Office. These forms will be used to create the Course Request data base. A management run will be produced to indicate conflicts, overloads, etc. After review by the academic departments and the Registrar's Office, the data base will be updated and the actual scheduling performed. During this same period, the student bills are prepared.

After the actual scheduling has been completed, the second management run will be printed and prepared for distribution. The Academic data base will be created to indicate the output of the scheduling. In addition, class tickets for the individual students will be printed. These are sent to the students approximately two weeks before the end of the current quarter. The student will report any conflicts to his advisor, where a terminal will be available for interface with the Academic data base. The conflicts will be resolved on the terminal and the data base updated. A new class ticket will be prepared and sent to the student. The final process in this phase would be the class roll, and the updating of any related systems, for example, Student Accounting.

Student Accounting Subsystem

The Student Accounting phase of the system basically follows a mid-quarter to mid-quarter cycle, and has as its main purposes the billing of the student for quarterly charges and the processing of the receipt of payment from the student.

The statement of charges sent to the student will not only reflect the charges which have been placed on a student's record, but also any credits which should be deducted from his bill. Since one of the major deductions to the student's fees is financial aid, one of the first steps in the system is to produce the necessary financial aid reports.

After all necessary corrections are made, the requested hours are used to calculate the charges. The bills will also contain any past charges not yet paid. A punched card will accompany the bill when it is sent to the student. This will be returned with the payment.

A student may make a payment in person at the Treasurer's Office or by mail in the time period remaining before the beginning of the new quarter. The payment program takes the receipt and distributes the amounts to the proper revenue codes. This provides input to the Virginia Tech Accounting System. The deferred payment plan and any other unusual conditions are handled by the system.

The last step in the Student Accounting is the recalculation of fees following adds and drops, and resignations. The proper data bases are updated, and the required reports furnished to the administration. Finally, the financial aid awards are transferred to the proper student records as cash payments.

Personnel/Payroll

The Personnel/Payroll system at Virginia Tech is in the process of being implemented. The system is logically segmented into two distinct parts which parallel the manual processes within the Virginia Tech Personnel Department. The Personnel segment concerns employee activity originating at Virginia Tech. The appropriate forms are constructed and transmitted to the State Division of Personnel, Richmond, Virginia. All payroll administrative activity origininating at the University and involving the State Comptroller's Office comprises the Payroll segment. Both segments are combined to form an integrated information processing system.

Personnel

Data is entered into the data base via a terminal located in the Virginia Tech Personnel Department. The data base contains all the necessary data for current operation, and has been designed to handle expected future needs concerning personnel data. The Personnel data base will have logical relationships with several other data bases in the system, such as Position Inventory, Class Code, and Wage Payroll. Ultimately, logical relationships will exist with data bases in other applications areas. The Personnel system is highlighted by the inclusion of many powerful features, some of which are listed below.

Computer-to-computer transmission of data
Modularity in design
High degree of security
Report generation capabilities
Rapid retrieval of personnel data

Payroll

The Payroll segment is designed to automate the payroll administrative processes that occur within the Virginia Tech Personnel

Department. The Personnel segment will interface very closely with this segment to be sure all the necessary data is available. It is an objective of the Payroll segment to contain and process all available and necessary data related to salaries and deductions for all employees (both faculty and classified) that are employed by Virginia Tech. As with the previous data base, the Payroll data base will have logical relationships with several other data bases in the system. There are several features which stand out in the Payroll system.

Automated balance procedures
Rapid retrieval of Payroll data
Accuracy
Minimum of manual administrative effort
Payroll distribution data
Maximum security

Since this system is aware of security measures, and has to have total compatibility with the state-wide system, a team approach was emphasized. Without a combined effort, the system cannot succeed; with it, the final product will become a valuable part of the Personnel Office's operation.

Other Productive Systems

Research Proposal Monitor System

The system maintains one data base which has a record for each proposal that is processed through the office of the Dean of Research.

The data base is updated regularly to insure accurate, up-to-date information on each proposal. Several monthly reports are produced, and the data is available for various on-demand reports.

The four main objectives outlined in the original proposal were accomplished when the system was implemented.

- Flexibility in use of research proposal data.
- Preparation of required University reports.
- Provision to the Research Division of a method of monitoring proposals that have been submitted, resulting in a better knowledge of the types of research and established obligations.
- Provision of a history, which can be used to help guide activity in both research planning and budgeting.

The system has been efficient since implementation, and provides useful data to systems in other application areas.

Gift Giving System

The Alumni, Student Aid and Educational Foundation Gift Giving System provides the University with a standardized method of reporting gifts received from various sources. The system provides an automated reporting method for gifts to the University. Information can be classified into various sequences for administrative reports. In addition, the system interfaces with the Alumni System to produce mailing information for those alumni who have not contributed during the calendar year. Actual mailing labels are produced for the Alumni Office.

The data base in this system contains three different types of information; Demographic Information, Giving History Information, and Current Year Description Information. The data base is updated by the Alumni personnel using an IBM magnetic tape selectric typewriter (MTST) and its tape cartridge.

Alumni System

This is one of the few administrative systems at Virginia Tech which is not implemented under the IMS data management system. The major functions of the system is to maintain, in a macnine readable form, biographic information on every living alumnus and friend of Virginia Polytechnic Institute and State University. This information is maintained on a sequential tape file in social security number order.

Input to the system is provided by the magnetic tape selectric typewriter (MTST). During the updating, a biographic visual card is produced for the Alumni Office. The system is capable of producing many lists based upon desired selection criteria.

The system is maintained entirely by personnel in the Alumni Office. In the future, the system may be redesigned for implementation under the data management system.

Space Utilization

A system is maintained by the Department of Physical Plant Planning to compute the University space utilization. This is another OS system and eventually will be converted to the data management system. The system maintains data on all rooms assigned to the University and their use during each quarter. Reports from the system are used by the administration to help determine the amount of funds to be requested from the Virginia General Assembly, and therefore are quite important to the University. Presently, punched cards and magnetic tape are used to operate the system.

This system can use data from several of the other systems, and can provide data useful in the other application areas. For that reason, it is high on the list for redesign and implementation under IMS.

FUTURE DEVELOPMENTS AT VIRGINIA TECH

The previous sections have defined systems that are in operation at Virginia Tech or in the process of being implemented. The goal of the Office of Systems Development is to develop a University wide management information system. In order to accomplish this goal, a long range plan was set for various application areas. This section will briefly define the systems that are (or soon will be) in some design phase.

Facilities

This system will basically replace the existing Space Utilization system. A record will be maintained on the data base for every facility assigned to the University. The root segment will contain all the general data relating to the facility. Different types of dependent segments will include overall space information, financial data, and some breakdown of the facility. The breakdown, for example, should be rooms for a building or spaces for a parking lot.

The data base will be updated semi-annually and will rely heavily on input from the University departments. Logical relationships will be established with data bases in several other application areas. The variety of reports will primarily depend on the demands that are required by the University Administration.

University Fixed Assets

An initial proposal, already presented for this system will be a method of keeping track of University equipment. This is extremely important in the Research area, where much of the equipment is purchased with federal funds. A method of estimating the value of the contents of a building can also be useful in case of disaster.

An Inventory Clerk, in the Accounting Office, will be responsible for keeping the system up-to-date. By interfacing with other systems and receiving required input from University department heads, the data base can be updated as required. Internal and external reports will be available to indicate the fixed assets possessed by the University.

University Book Store

A study of the University Book Store indicates an automated inventory is applicable. Such a system will include not only books but supplies. The combination of an automated inventory system and a solution to the book-demand procedure would solve the stock-out problem that frequently exists. The solution to the demand problem would be the development of a forecasting procedure which could result in logical relationships with other data bases.

Another portion of this system would be the entry of information at the point of sale. The inventory would be reduced at that time and the sale recorded. Daily reports would provide the Book Store with information needed to prevent the stock-out.

Purchasing

The Purchasing Department is divided into several areas: 1) Central Stores, 2) Scientific and Electronic Equipment, 3) Industrial Furniture, and 4) Food Stores. The Purchasing Department acts as a service division to the University, issuing purchase orders by means of purchase requistions from each department.

Inventory systems should be proposed for each of the areas indicated above. Then there should be an interface with those inventory systems to provide the University with an up-to-date inventory system on an institution-wide basis. Such a tie-in would permit various departments to coordinate more efficiently their inventory needs through the Purchasing Department. In addition, an automated inventory system would aid in providing turnover rates of items, could feasibly lower storage requirements and additionally, would provide back-up procedures under emergency situations. A more detailed study in this area should bring to view other considerations such as:

Central receiving
Computer generated Purchase Orders
Automatic verificaton of available funds
Use of historical data for evaluation
Forecasting techniques

The development of such a system would fill the many existing gaps in the University-wide management information system. As previously mentioned, offsprings of this system would be individual inventory systems for areas within the Purchasing Department.

Test Creation and Evaluation

This system has already been proposed. It will enable the faculty to have the computer prepare examinations, score the answer sheets, and evaluate the results.

Graduate School System

After a person has been admitted to graduate school, this system will act as a monitor. The functions will be: to monitor an individual's program of study; to review the status of Graduate Teaching Assistants and Graduate Research Assistants, and to keep track of many scholarships.

Other Systems for Consideration

- Food Services
- Building and Grounds
    - Carpentry Schedules
    - Custodial Services
    - Mail Service
- Motor Pool
- Utility Departments
    - Power Plant
    - Plumbing
    - Security
- Information Services
- Extension Division

Some of the systems named in this section have already had an initial study, while others have been mentioned for future studies.

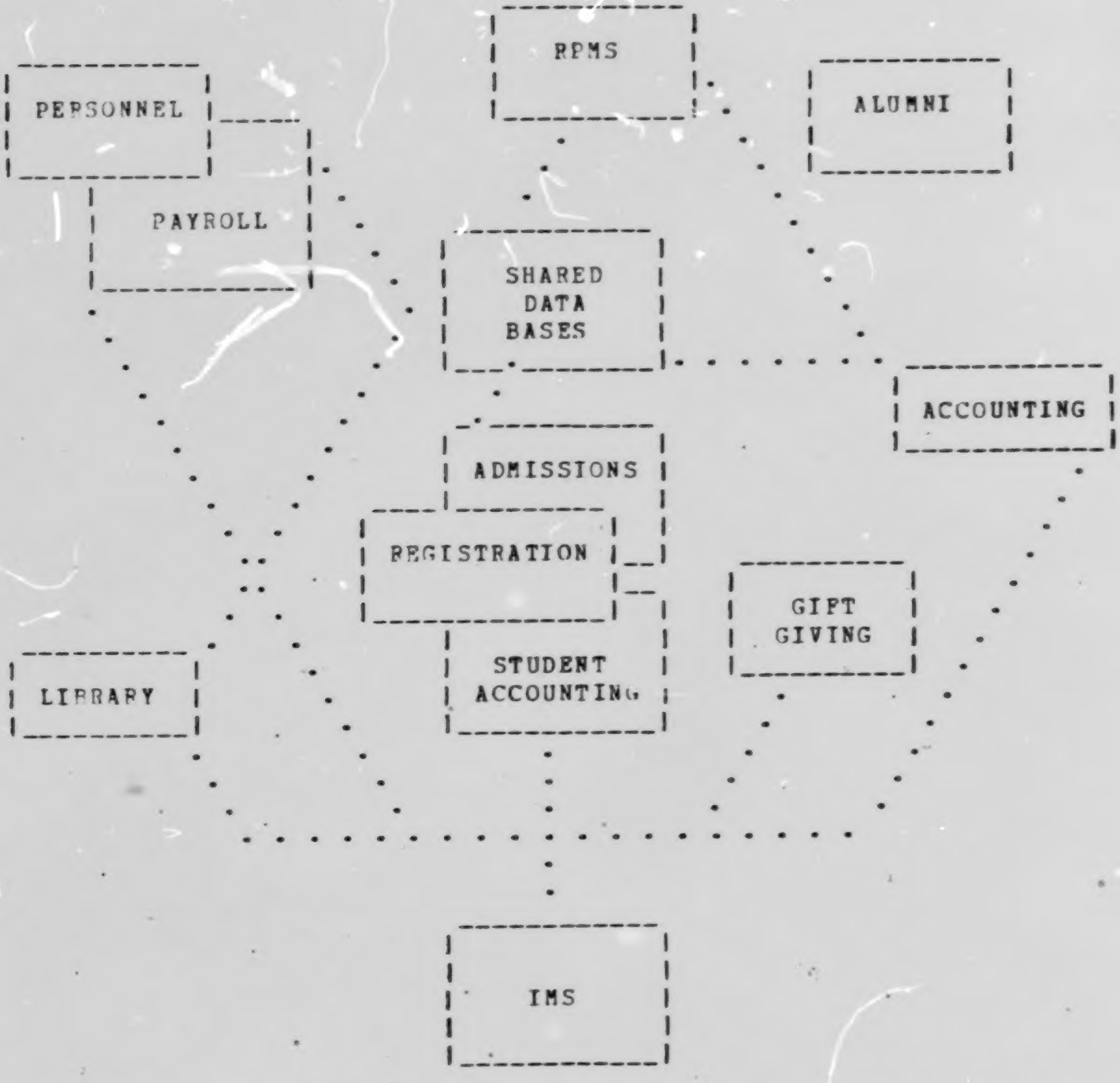
Several of the systems will be inter-related. It is not necessary to have all these systems implemented to provide a University-wide management information system. How the Office of Systems Development plans to achieve this system is discussed next.

APPLICATION INTEGRATION

When the Office of Systems Development was established it set several goals. The primary goal was to develop a University-wide management information system within 5-7 years. The previous sections of this paper have shown the organization of Systems Development, the reasons for selecting IMS, the procedures for designing a system, and the different application areas at Virginia Tech. The purpose of this final section is to present the plan for accomplishing our primary goal.

Present Position

Figure 6.1 indicates the systems that are in operation or being implemented at Virginia Tech. It can be seen that relationships exist within application areas. For example, within the area of Student Records, there are logical relationships between the various systems. In some cases, an area, such as Student Records, may provide input to another area, such as Accounting. At the present time this is a manual procedure. We can summarize by saying that the present degree of integration is limited by systems within an application area. To take advantage of the IMS features, several data bases were established to be shared by the various areas. These are mostly in the form of tables, but they do reduce the occurrence of redundant data.


RPMS
PERSONNEL
ALUMNI
PAYROLL
SHARED DATA BASES
ACCOUNTING
ADMISSIONS
REGISTRATION
GIFT GIVING
LIBRARY
STUDENT ACCOUNTING
IMS

Present Position of Application Integration
Figure 6.1

Integration Plan

Hopefully, it is now evident why Systems Development took many of the actions previously defined. IMS was selected because of the data base features. The department was organized having project managers for different areas, but still having a structure where everyone works together. The various committees and the procedures for systems design supported the organizational structure and established easy communication within the group.

A sophisticated management information system is extremely difficult to achieve. Systems Development took the approach that the first step should be a simplified system. Certain goals should be set and achieved with this first system before proceeding to the total system. Specifically, the first system should sell the user by providing him with benefits not previously available.

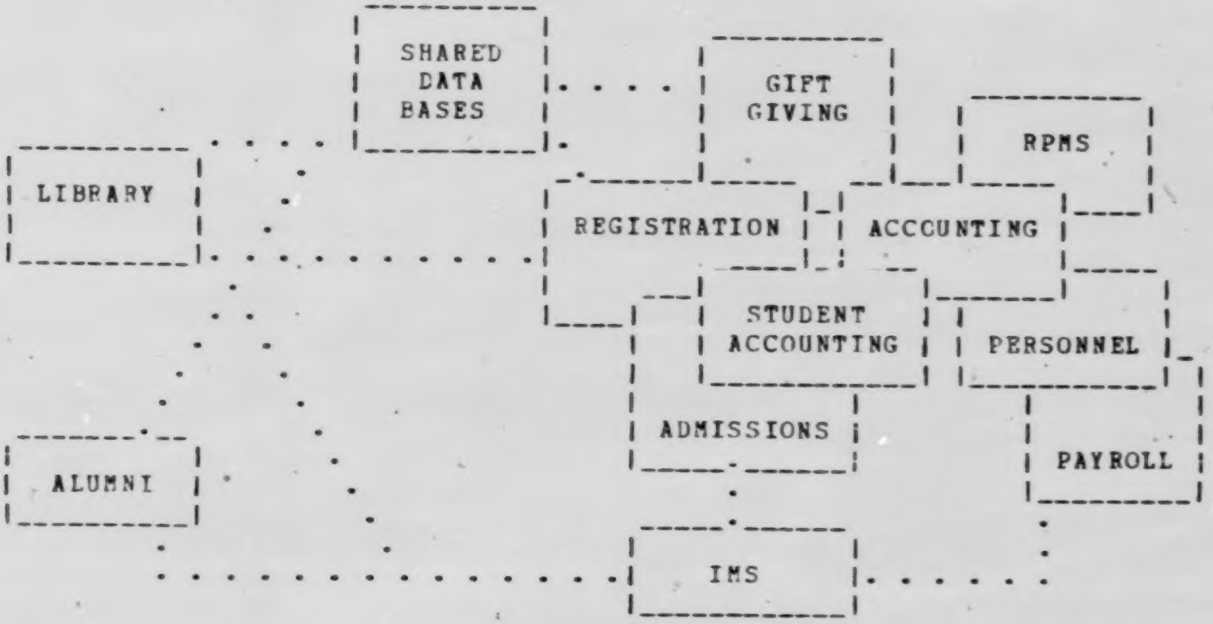
The plan at Virginia Tech was to develop systems in the various application areas and accomplish the above-defined goals. The areas have not been integrated except for the use of shared data bases -- a step to eliminate redundant data. Previous sections have defined the systems presently in operation at Virginia Tech. In most cases, these systems have accomplished the three initial goals. In addition, because of the development of certain systems, terminals are now located in each college within the University and in several administrative offices. Systems Development can now successfully begin the integration of the various application areas. Some of the new areas to be developed will immediately fit into the integration, while others will go through a process similiar to the initial ones.

Accomplishment of Application Integration

The Project Managers have worked together closely during the past few years knowing that application integration was on the way. Through the various committees, standards were adopted in many areas. In the data base committee, the structure, contents, and access methods were discussed in detail by the Project Managers. Preparation for application integration has been in process for several years.

With the increase in the number of systems being implemented at Virginia Tech, logical relationships between data bases in different application areas have become realistic. In some cases logical relationships are not feasible, but an interface is established. The Data Base Administrator has the final say as to the degree of integration.

Figure 6.2 shows the application integration which will be implemented within the next year.

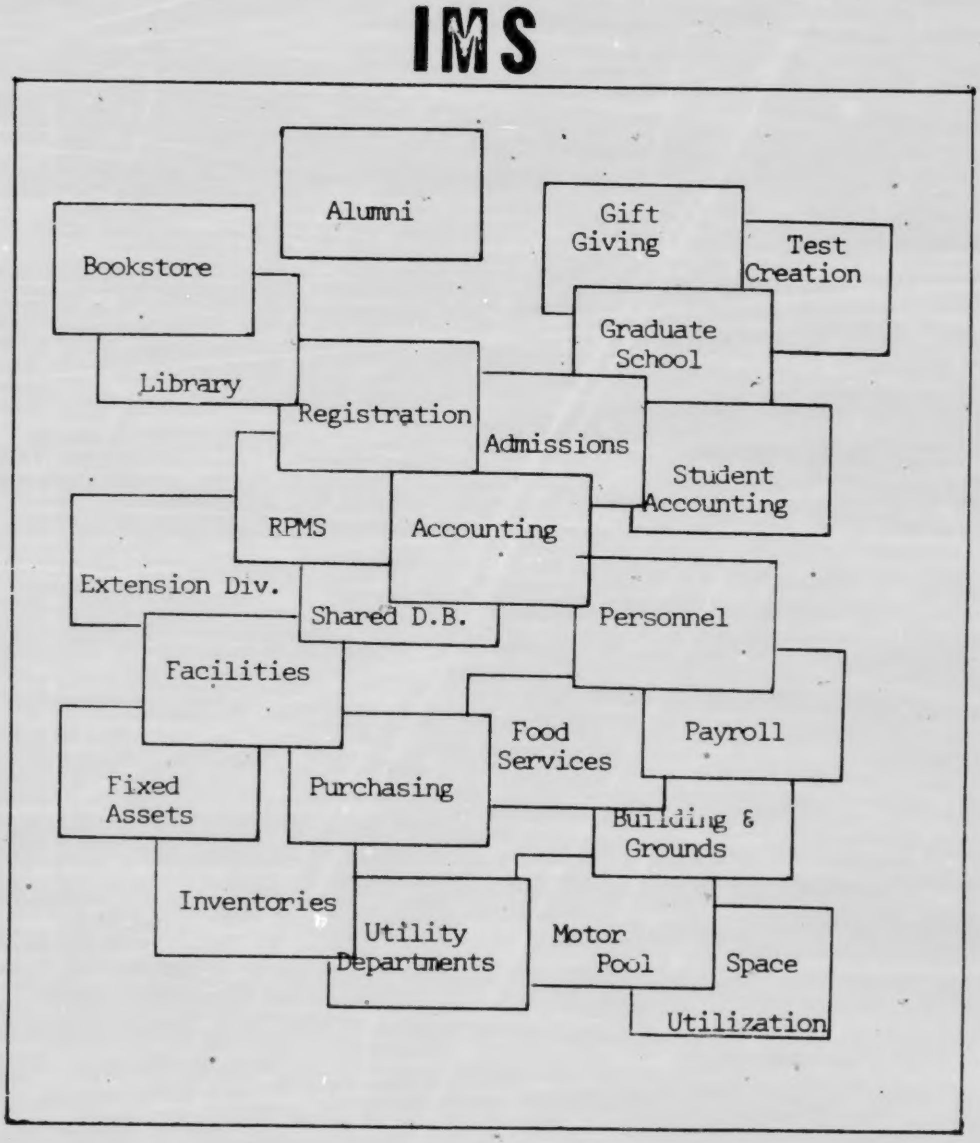

SHARED
DATA
BASES
GIFT
GIVING
RPMS
LIBRARY
REGISTRATION
ACCOUNTING
STUDENT
ACCOUNTING
PERSONNEL
ADMISSIONS
ALUMNI
PAYROLL
IMS

Application Integration Within Year
Figure 6.2

The implementation of the Personnel data base will be the initial step toward the integration. The existing Accounting and Student Records areas will utilize data in the system for several administrative reports that are now a manual process. In addition, the terminals located in the colleges will have the capability to inquire on data from most of the application areas. For example, accounting, and registration data is presently available, and the personnel data will be available in early 1974. As more systems are implemented, additional information will be accessible.

Future Plan

The Office of Systems Development will continue its schedule for developing the University-wide management information system. We are hopeful that terminal usage will be expanded to the individual departments. The simple system has been implemented and has accomplished its goals. The initial step for the total system has been taken, and it too has to accomplish the goals. It will be the responsibility of Systems Development and the users to continue the close teamwork, and to fulfill the goal of the successful University-wide management information system (Figure 6.3).


IMS
Alumni
Gift Giving
Test Creation
Bookstore
Graduate School
Library
Registration
Admissions
Student Accounting
RPMS
Accounting
Extension Div.
Shared D.B.
Personnel
Facilities
Food Services
Payroll
Fixed Assets
Purchasing
Building & Grounds
Inventories
Utility Departments
Motor Pool
Space Utilization

University-wide Management Information System

Figure 6.3

DESIGNING LARGE NETWORKS

Craig Johnson
Cambridge Scientific Center
IBM Corporation
Cambridge, Massachusetts

DESIGNING LARGE NETWORKS

With state-wide, regional, and consortia activities in the area of computer services to institutions of higher education, the design of large information networks is of increasing interest to administrative data processors. This presentation presented the salient characteristics of design for large information networks.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

AN EDUCATIONAL COMPUTER USERS NETWORK

Thomas H. Beerman - Gregory S. Jonesku
Data Management and Computer Graphics
Battelle Columbus Laboratories
Columbus, Ohio

BECUN
THE EDUCATIONAL COMPUTER USER'S NETWORK AT BATTELLE

Thomas H. Beerman
Gregory S. Jonesku

BATTELLE
Columbus Laboratories

The Educational Computer User's Network at Battelle represents a unique application of the joint-computer-center concept. The program is unique in two respects. First, it is a cooperative effort between a group of Ohio colleges, secondary schools, and a large research-oriented organization. Second, the entire program has proceeded without any federal or other outside financial support. Both Battelle and the schools have participated by utilizing funds available within their own operating budgets.

The purpose of this paper is to describe the program and some of its results from the viewpoint of Battelle's Computer Center.

HISTORICAL BACKGROUND

The Columbus Laboratories of Battelle comprise the original research center of a public-purpose, multinational organization devoted to scientific research and development. Battelle's broad objective is the advancement and utilization of science for the benefit of mankind through the processes of technological innovation, and the better education of man. It was founded in 1929 in Columbus, Ohio, by the will of Mr. Gordon Battelle, an Ohio iron and steel industrialist.

Battelle-Columbus acquired its first digital computer--an IBM 650--in 1956. For several years this computer was used primarily in connection with one large research project. However, beginning in 1960, two much broader objectives were established for the Computer Center. The first objective was to encourage and support data processing throughout the Battelle-Columbus Laboratories. The second objective called for the Battelle Computer Center to become a leader in data processing research. To help achieve these objectives, an active program was undertaken to encourage other operating divisions within Battelle-Columbus to learn how to use computers in their fields of technology. In addition, the IBM 650 was replaced with a more modern, state-of-the-art computer, and people with broad industrial and military experience in data processing were brought to Battelle.

During the next five or six years, computer technology advanced rapidly. By 1966 the Computer Center was already considering a replacement for its third computer system. Computer science graduates were beginning to appear for the first time and people with strong technical backgrounds in data processing were much easier to find. Unfortunately, however, graduates of small liberal arts colleges with exposure to data processing were almost impossible to locate. The reason was that most such colleges had simply not been able to keep pace with this new field of technology. Not only had their graduates usually no significant exposure to computer technology, but in many cases even the basic terminology of computing was unknown to them.

At about this time, a group of seven Ohio colleges recognized the problem and decided to do something about it. They banded together in the hopes of finding a mutual solution to their common problem. As a first step, the group jointly sponsored a systems study to determine the extent to which data processing capabilities were needed by the colleges and to identify means of acquiring these capabilities. This study, conducted for the colleges by an outside organization, recommended that the colleges undertake a cooperative program in computing and listed three possible sources for a centralized computer facility: (1) a new computer center jointly established and supported by the seven colleges; (2) The Ohio State University; and (3) Battelle's Columbus Laboratories. After considering the recommendations of the report and other factors, the seven colleges decided to approach Battelle for the purpose of developing a joint proposal to the National Science Foundation for financial aid.

An NSF proposal was subsequently developed by the seven colleges and Battelle-Columbus. The proposal called for a three-year program. Under the program, the existing Battelle-Columbus Computer Center would serve as the central facility and provide the focal point needed to mount a cooperative effort in data processing. Each college would have access to the Computer Center facility via a remote batch terminal. In addition, the proposal covered such areas as staff acquisition, staff training, curriculum development, and evaluation of progress under the program. The area of administrative data processing, however, was considered to be outside the scope of the proposal.

After many review cycles, this proposal was put into final form and submitted to the colleges for signatures. Unfortunately, one of the seven schools had, in the meantime, decided to go its own way in computing and declined to approve the proposal.

At this point, more than a year's effort seemed in danger of being wasted. However, Dr. Hans Jenny of The College of Wooster came to the rescue. Dr. Jenny decided that Wooster must have a data processing capability, and proposed that if Battelle-Columbus were willing to go ahead with its portion of the program, The College of Wooster would participate without NSF support. This decision marked the beginning of the Battelle-Columbus College Program.

By December of 1967, The College of Wooster and Battelle-Columbus had entered into a contractual arrangement. For the next six months the use of the computer by Wooster remained sparse, as Wooster personnel had to commute back and forth to Columbus, a distance of 100 miles. However, in the summer of 1968, Wooster acquired a remote batch terminal. Computer usage then picked up dramatically with the start of the fall term and has grown steadily ever since. Later, during the summer of 1971, multiplexing hardware was added to Wooster's communication system and with the start of the fall term in 1971 they had four interactive ports into the Battelle computer system.

As the news of Wooster's success spread, other colleges became interested. During the summer of 1969 both Otterbein College and Antioch College joined the program. (Otterbein and Antioch, like Wooster, had been members of the original seven college group.) At the present time, both of these schools are utilizing remote batch terminals; Otterbein is also doing interactive computing via teletype. A fourth school, Ohio Dominican College, joined the program late in 1971. As yet, most of Ohio Dominican's use is via courier and a teletype. Two schools, Capital University and Columbus Technical Institute, joined in 1972 beginning with interactive processing via TTY's. Capital is currently installing a remote batch terminal.

Also in 1972, a move was made to further strengthen the program. The Center for Improved Education and the Computer Center at Battelle combined together to organize all educational computing activities, including colleges and high schools, into one group--known as BECUN. The Center for Improved Education's primary mission is to help schools and colleges design their education systems through the use of existing knowledge. The Computer Center provides technical expertise, the software, and the hardware.

BECUN currently consists of six colleges and ten high schools. The remainder of the paper will cover those aspects concerning primarily the colleges.

PROGRAM CONCEPT

Under the program, all of the resources of the Battelle-Columbus Computer Center are made available to the colleges. This means that they are provided access to the same hardware, software, and technical assistance as Battelle's own large staff. The colleges are, in effect, treated as an extension of the staff in this respect. Routine technical assistance is made available free of charge and there is an absence of discrimination in matters of usage, charges, turn around, and so on. These policies enable the colleges to feel they are indeed "first class" users. Further, the colleges are free to use the Battelle facilities as little or as much as they wish, paying only for what they use when they use it. Membership in the program does not commit the colleges to any long-term arrangement, and they can use other sources of data processing as they see fit. Finally, cooperation among the members is actively encouraged and supported--this being made a good deal easier, of course, by the use of a common facility.

All of this is consistent with the objective of the program, which is to assist the colleges in achieving a mature posture in the use of computing for their academic, administrative, and research needs. The intention is to perpetuate the program only so long as it best serves the interests of the colleges. Thus, when the members--either independently or collectively--decide their purposes would be better served by a different course, Battelle expects to cooperate fully in whatever way is appropriate.

DEVELOPMENT UNDER THE PROGRAM

In terms of the growth pattern in data processing at the colleges, the program can be thought of in terms of four phases. During the first phase, a college typically uses couriers to bring work to Battelle. Often this marks the college's first exposure to data processing. Ohio Dominican--located in Columbus--began their activity in this manner. Once a day a student would bring work to Battelle and wait for the output; usually his wait would be less than a half hour. Otterbein, which is located near Columbus, made two trips a day for about one year when in this phase. Antioch, which is 55 miles from Columbus, made three trips per week. Wooster's travel during this phase was on an as-needed basis.

As one would expect, the growth in computer usage is comparatively slow during this first phase. However, this phase does provide time for the colleges to develop administrative applications so that when they do acquire a terminal they will already have some existing workload.

The second phase begins with the acquisition of a remote batch terminal. Wooster, Otterbein, Antioch, and Capital all currently have Control Data Corporation 200 User Terminals. These terminals consist of a 330 card-per-minute card reader, a 300 line-per-minute printer, plus a keyboard and cathode ray tube. The terminals are linked via telephone lines to the Battelle Computer Center. During this phase, the colleges operate the terminal as though it were a stand-alone computer. The same terminal is used for academic, administrative, and research purposes. Operational procedures resemble those seen in many other open-shop computer facilities.

During this phase, growth is usually rapid in all areas. Additional administrative packages are developed and implemented. Students and faculty quickly discover the capability of the "computer" and put it to good use. Usually, at the conclusion of this phase the colleges have already developed a very sophisticated batch-processing capability.

The third phase is very closely related to the second and begins when interactive computing capability is added. With this capability, the student and the researcher now have the choice of using either the batch or the interactive approach to applications. The administrative systems designer can utilize this capability for source data entry or for on-line data retrieval from immediately accessible data bases. The system in use at Battelle permits the user to move with complete freedom between the batch and interactive environment, choosing whichever best suits his purposes at the moment. No program conversion is required since the same software is used in both environments. Also, a data base developed in one environment is accessible from the other.

At the present time, both Wooster and Otterbein have entered this phase. Otterbein's use of interactive computing started in January, 1972. Wooster, which had some previous experience with interactive computing, began using the Battelle facility for all their work during the summer of 1971. Antioch, for valid reasons of their own, does not presently support interactive computing.

The fourth phase is one that we expect to enter during 1974. This will consist of developing and making available capabilities for Computer-Assisted Instruction. To support CAI, Battelle and the member colleges have been following the development of a CAI system and will be implementing it during 1974.

HARDWARE AND SOFTWARE

Tables 1 and 2 show the hardware that is currently installed.

As shown in the tables, the Computer Center operates both a CDC 6400 and a CDC CYBER 73-16. These computers are roughly equivalent in internal speed and both are in the large-computer category. The 6400 has 655,360 characters of central memory, while the CYBER has 983,040 characters. One machine--the 6400-- is devoted to interactive computing, while the other handles local and remote batch processing. The two computers are tied together by a 5.12 million character large core memory and communicate via this means.

The interactive computer has 81 communication ports. The communication speeds vary from 110 baud to 300 baud asynchronously, and from

TABLE 1. BATTELLE-COLUMBUS COMPUTER CENTER HARDWARE AS OF JANUARY 1, 1974

Quantity	Device	Characteristics
	TIMESHARING HARDWARE	
	Computer	
1	CDC 6400	1 million instructions per second
12	Data Channels	
10	Peripheral Processors	
1	Central Memory Unit	655,360 characters
3	Communication Multiplexers	Total of 81 ports
	Mass Storage	
1	Large Core Memory	2.56 million characters
13	Disk Pack Drives	110 million characters each
	BATCH HARDWARE	
	Computer	
1	CDC CYBER 73-16	1.2 million instructions per second
12	Data Channels	
10	Peripheral Processors	
1	Central Memory Unit	983,040 characters
1	Communications Multiplexer	16 ports

BATCH HARDWARE
(Continued)

Quantity	Device	Characteristics
	Mass Storage	
1	Large Core Memory	2.56 million characters
12	Disk Pack Drives	8 of 35 million characters each 4 of 6.25 million characters each
1	System Disk	176 million characters
10	Magnetic Tape Drives	7-track
	Input/Output Equipment	
3	Card Readers	1200 cards per minute
5	Line Printers	2 of 1200 lines per minute 2 of 1000 lines per minute 1 of 500 lines per minute
1	Electromechanical Plotter	11-inch drum
1	Card Punch	250 cards per minute
	MICROFILM RECORDER	
1	Stromberg DatagraphiX 4060 Microfilm Recorder	16mm or 35mm sprocketed, 16mm unsprocketed cameras. Immediate positive film and hardcopy for 35mm output
	Includes	
1	Honeywell 516	Half-million instructions per second, real-time clock
1	Central Memory Unit	16,000 16-bit words
2	Magnetic Tape Drives	1 7-track 1 9-track

MICROFILM RECORDER
(Continued)

Quantity	Device	Characteristics
1	Analog-to-Digital Converter	
1	Paper Tape Reader/Punch	300 characters second reading 110 characters per second punching
1	Mark-Sense Card Reader	100 cards per minute

CYBER GRAPHICS TERMINAL

Quantity	Device	Characteristics
1	CDC CYBER 777 Graphics Terminal	24-inch refresh tube, lightpen, hardware windowing and scissoring

Includes

Quantity	Device	Characteristics
1	CDC SC-1700	Half-million instructions per second
1	Central Memory Unit	32,000 16-bit words
1	Disk Pack Drive	6.25 million characters
1	Communications Controller	Communicates with CDC 6400 at 40.8-50 Kilobaud
1	Card Reader	300 cards per minute
1	Line Printer	300 lines per minute

TABLE 2. OTHER HARDWARE (TERMINALS)

Quantity	Device
1	Data 100 Model 72 Remote Batch Terminal
2	Tektronix 4012 Direct View Storage Tube (DVST) Graphics Terminals
1	Computek 400/15 DVST Graphics Terminal
1	Texas Instruments Model 733 ARS (with dual magnetic tape cassettes)
1	Texas Instruments Model 733 KSR
1	Texas Instruments Model 725 KSR (Suitcase portable)
12	Model 33 Teletypewriters

2000 baud to 40.8 kilobaud synchronously. Twelve large disk pack drives, storing more than 100 million characters per drive, are available for on-line storage of data.

On the batch computer there are 16 communication lines in the speed range of 2000 to 4800 baud synchronously. There are also 10 tape drives available to both the local and remote batch user. Three types of disks are available. These consist of a large disk file for tape simulation and scratch files, plus four small (6.25 million character) and two large (35 million character) disk pack drives for use as private devices. The latter are used in a way similar to magnetic tapes but can be randomly accessed. In addition, the user has access to 210 million characters of on-line disk storage and 800,000 characters of large core memory.

All of the various printers, plotters, punches, and paper tape devices located at the Center are directly available to the remote user. Output from these devices is ordinarily mailed to the colleges. The same is true for the large, high-speed off-line microfilm plotter. All of these devices have been used productively by Otterbein, Antioch, and Wooster.

As regards software, both the interactive and remote batch user can access all of the Computer Center software. The languages in general use by the colleges are BASIC, FORTRAN, COBOL, and SNOBOL, and they use a wide range of application packages as well. Both the Dartmouth and Beloit libraries have been implemented.

Much of the general-purpose software development--or conversion of packages acquired elsewhere--has been a joint effort between Battelle and one or more of the colleges. In some cases, Battelle has supplied machine time at no cost with the understanding that the resulting software would be made available to all Computer Center users.

One important function of the Battelle Computer Center is performed by the Advisory Services Group. The mission of this group is to assist users with programming problems and to provide general coordination of user activities. This type of assistance is available to the colleges at no cost.

Each of the colleges perform a similar function for their users. In general, this is done through the appointment of a "coordinator of data processing" at the college. The coordinator handles all communication between the college and Battelle. He concerns himself with policy problems, budgets, operations, library functions, and consulting. All information coming to or from Battelle passes between him and his counterpart in the Battelle Advisory Services Group. The outstanding caliber of these college coordinators has been one of the major keystones in the success of the program.

Beyond the normal day-to-day need for assistance, Battelle-Columbus has provided staff assistance to the colleges on a cost-incurred basis.

Battelle personnel have, for instance, taught courses at each of the colleges. An administrative systems study and design was performed for Otterbein, which later implemented the design. As another example, a joint effort between Antioch, Otterbein, Wooster, and Battelle led to the design and implementation of a sophisticated alumni history and gift accounting system. Many other smaller projects have been carried out jointly.

We at Battelle believe that the colleges participating in this program have access to as much data processing power as many of the largest universities. At first exposure to this power, each college has a natural tendency to flounder a bit, wondering where to start. However, with the advice and help of Battelle and the older members of the group, each new member has been able to move ahead quickly and to make rapid progress toward achieving maturity in data processing.

COSTS

Under the program, every job is costed and billed to the college. Therefore, the college user must decide, before making a run, whether the use of the computer is justifiable. With this approach, application programs are not developed just "because there is extra computer time and thus there is no cost". Students learn that the computer is a tool--not a toy--because their grade may depend upon their computer costs. Sloppy systems design and programming are not easily hidden. In far too many in-house computer facilities, applications are developed simply because the machine is relatively idle. Users come to depend upon these applications. Later, as the computer load grows, it becomes obvious that some of these "free" applications are not cost-justifiable, but it is then probably too late to get rid of them. As a result, these applications are reflected in increased hardware costs for expansion or a larger machine. In the present program these problems are avoided because the user is forced to consider economic justification at the start.

The members of BECUN pay the same rates as all other Battelle Computer Center users. The schools are, however, treated as a group sponsor (a group of individual concerns jointly sponsoring a single research activity). This allows the group to qualify for bulk-use rates. In fact, after an initial base-usage level is reached each month, costs are discounted up to 80 percent. This, of course, does much to encourage long-term growth in computing at the colleges.

During its first year of participation in the program, each school is guaranteed that its computer usage costs will not exceed $1500 per month. This enables colleges entering the program with little or no computing experience on other systems to experiment. It also helps colleges with experience on other systems to convert their existing applications. Further, it insures any new college that the costs of learning a new system will be tightly controlled, and permits the college to gather information to project their data processing budget for future years.

In the Battelle Computer Center's rate structure, each resource is priced separately. The user can choose those resources which solve his problem at least cost. He can, for example, choose between sequential tape files and random or index-sequential disk files. At varying costs, his data can be stored on-line, or in on-site or off-site storage areas. His applications can operate in a batch or interactive environment, or both. He can produce printed, graphical, or microfilm output. He can use a particular resource once a year or many times a day. In addition, he does not have to be concerned with justifying the acquisition of new hardware to meet the need of "that new" applications program.

In today's unbundled environment, software packages can cost money too. One of the policies of the Computer Center is that all software developed or acquired by the Center is made available to all users. Because of this, the colleges have a wide choice of computer languages and application packages available to them. Frequently software products are needed at a college for one course or part of a course, but the software license may require a minimum lease period. Individual justification would be difficult; however, with six colleges plus Battelle's internal needs, such packages are often easily justified.

How do the costs of participating in the program compare to having a typical small in-house computer? Assuming that a college has had little or no previous data processing experience and looking at a five-year comparison, the total costs over the first four years are roughly equivalent for the "average" college. If there is a cost advantage, it would appear to be in favor of the program. If the college is one that progresses more rapidly than the average, there is a distinct advantage to the program, whereas, if the college progresses more slowly then the advantage will shift depending upon whether progress was comparatively slower during the "first" or "second" year.

The big advantage of the program occurs after the fourth year. By this time the college has outgrown the small computer and finds that the next step upward in hardware is quite expensive. In addition to the increase in monthly rental, there will be conversion costs if the language of the small computer is not supported on its replacement. However, the most distressing aspect is likely to be that there is no one practical choice of a computer that will provide the needed additional capability for both administrative and academic users. The needs of one or the other must be sacrificed or two new computers will be required. None of this happens under the program, of course--costs simply continue to increase modestly as usage grows.

There are other cost factors which favor the program. Battelle and the colleges have jointly purchased supplies, thus reducing these costs. In-house software support, of course, is less; for example, the colleges do not need to hire systems programmers or contract with the manufacturer for software support. Also, the broad base of experience existing at the colleges and Battelle can frequently result in immediate answers to questions that would otherwise require experimentation and create delays.

What about cost control? When talking to prospective new members, one of the most frequent fears we hear is that they will be unable to control their costs. There is a general feeling that with an in-house computer, monthly costs are fixed. And perhaps they are--until one needs to upgrade! Neither students, faculty, nor administrative personnel will be long satisfied with the argument "that the computer is loaded". When pressures mount, a large incremental cost instead of a gradual increase in cost then becomes the problem.

It is possible to exercise control and the colleges do do it. In each college, the data processing center acts as a cost center. They pass both Battelle's costs and their own center's costs along to the end user. Battelle costs are available to them on a daily basis. Each day--as they start operation--the month-to-date costs on each account, plus individual job accounting, are available to them through the computer on an on-line file. On some regular basis, generally weekly, they process this data by running their own accounting program. Thus the end user, as well as the Data Processing Director, can be constantly aware of his costs.

As indicated at the beginning of this section, job-by-job accounting and costing can be a very effective tool for cost control. Nevertheless, it is often resisted. Frequently, we hear the comment that "it will destory the experimentation needed in the educational process". A similar claim is often made with respect to research. However, both research and the experimentation that goes with it have flourished at Battelle. The computer, after all, is a tool--and a relatively expensive one. Too much experimentation by a few can deny the use of that tool to others. The only effective control is cost--or perhaps we should say "cost visibility". Making the cost of each user (and of each use of the computer) visible is a useful function, regardless of who actually pays for or supports that cost. In general, if experimentation is well planned with reasonable objectives, it can be justified. In any case, simply ignoring that a cost does attach to experimentation--even with an in-house computer--serves few desirable ends.

ADVANTAGES OF THE PROGRAM

At this point, we will summarize the main advantages of participation in BECUN while also indicating many of the results which have been generated. Some of these advantages would be enjoyed in any effective program of facility-sharing. Others are unique to this particular program and stem partly from the need to satisfy Battelle's own research needs. The reader will be able to distinguish between the two. No particular significance should be attached to the order in which the advantages are listed.

(1) Cooperation Among the Participants--In the long run, we believe this will be one of the major benefits of the program. To insure regular contact, representatives from each college meet with Battelle personnel once a month. In the early meetings, most of discussion was between the colleges and Battelle. In the more recent meetings, the exchange of information has been primarily among the colleges themselves. A wide variety of experiences, needs, and desires are discussed in these meetings, and the rate of growth of each individual college is greatly enhanced by these exchanges. Cooperation has occurred in many other ways. Representatives from schools often visit each other to discuss specific application areas. The schools have also worked together in joint projects. A major example of this was the development of slide/cassette courses in the BASIC and COBOL programming languages. Under the primary direction of two professors from Otterbein College, the courses were developed with both financial support and technical review from all of the schools and Battelle.

(2) Hardware and Software Capabilities on a Par With the Largest Universities--This benefit is, for a group of this size, unique to the program. By the very nature of Battelle's research needs, the Computer Center must have large-scale computing capability current with the state of the art. The colleges reap the benefits. The monthly rental cost of the hardware in the Computer Center has more than tripled since the start of the program. We have heard it argued that the colleges do not need all of these capabilities. We agree that they do not need them all of the time. Nevertheless, each of the three older participants have, at one time or another, made good use of practically every capability available. And since the participants pay for a capability only when they use it, the mere availability of an "unnecessary" capability costs the colleges nothing.

(3) Costs Increase Gradually Rather Than in Large Increments--The benefits in this respect are obvious to anyone who has attempted to justify additional hardware or software. It is true that even though the cost of computer usage increases gradually, the colleges can be faced with some modest incremental increases. These occur when a remote batch terminal is installed, or when additional communications capabilities are added. While they may be as difficult to justify as any other increase in hardware costs, they do, fortunately, occur infrequently and are normally small in magnitude.

(4) The Colleges Receive a Service--As a group, the colleges represent one of the Computer Center's largest users. As such, they are in a position to request and support the services required to fulfill their needs. For example, the operational hours of the Computer Center have been changed at the colleges' request (Saturday hours were expanded and Sunday hours were added). Joint software development efforts sponsored by the group were mentioned in an earlier section and, although we have yet to acquire any major hardware specifically for the colleges, we have made software acquisitions for this purpose. In general, we feel that the colleges are availed of services for educational purposes far in excess of those of all but a few in-house university computer centers.

(5) Students Learn to Use the Computer as a Tool--With the available hardware and software, students can turn their attention to problem-solving as opposed to spending time "shoe-horning" a large application into a small machine. In addition, students learn how to communicate with the computer via a modern operating system as opposed to hands-on operation. This is an important point since most installations do not permit programmers hands-on utilization of the computer. In fact, in many installations the programmers seldom see the computer so that students trained in a hands-on environment must often be retrained at a cost to the employer. Students graduating from participation in BECUN learn to use data processing under conditions generally found in the outside world.

(6) Applications Are Economically Justified at Their Inception--This point was discussed in detail earlier. The main advantage is that costs for each and every application stay "visible" and so are subject to control.

(7) Staff Support--As discussed earlier, the Battelle staff has assisted the colleges in many respects. To date, almost all of this support has come from the Computer Center staff; however, future support could well involve other areas of expertise.

(8) Cost--This point was discussed in detail earlier. Both short-term and long-term savings can accrue to the participants.

PROBLEM AREAS

As can be expected with any major program applying new concepts and covering a variety of participants, problems have occurred. The following problems have been encountered at one stage or another and the appropriate actions taken.

(1) Communications--The distance between the central site (Battelle) and the remote sites (colleges) has made it more difficult to communicate appropriate information. It can be very frustrating to be at the end of a communications link wondering if the computer is down or if your job has been run. To overcome this, several actions have been taken. First, the monthly meeting between all participants insures an open forum for appropriate discussions. Second, documentation such as the Remote Terminal Information Bulletin is sent periodically to each school. Third, the schools often make trips to Battelle to discuss specific needs. Finally, efforts have been initiated to visit each school on a periodic basis as a form of circuit riding. These trips have often included Battelle staff from both Advisory Services and operations so that they may meet the people from the "other end of the telephone line" and relate better to their needs.

(2) Uncontrolled Usage--Since the computer's capacity is relatively "open-ended", its usage can be abused if completely ignored. It is sometimes tempting to let the computer do all of the work. Control of usage has been primarily implemented through complete and up-to-date reporting of all machine usage. A college's Data Processing Director can easily identify and review his usage including job-by-job accounting. In addition, special software has been implemented to permit limits to be established for the generation of on-line files.

(3) Own Equipment Syndrome--Since many people come from backgrounds of "hands-on" experience with their own equipment, they often hesitate at the thought of losing control of the resources. If they can be convinced to try connecting to a central site, experience has shown that they soon will overcome this feeling. Many of the current schools have indicated that they prefer not worrying about putting up a new operating system or having maintenance done on the central processor and can better devote their time to more important problems.

(4) Academic/Administrative Imbalance--During the first few years of operating a remote batch terminal, growth is normally more rapid in administrative computing. Thus, both input/output volumes and costs soon exceed those of academic computing. At this point it is important that the faculty/student needs are not overshadowed. Full cooperation is necessary in areas of scheduling, budgeting, etc. The colleges have handled this most successfully under the direction of the Data Processing Directors who must keep the needs of both in mind.

FUTURE PLANS

Many new areas for applying computers are continually being explored at each of the schools. In addition, new capabilities/techniques are being implemented. Otterbein has obtained an intelligent, interactive terminal which may be used off-line for data entry/edit in many administrative areas. The terminal can then be used to transmit the data to the central site for further processing.

Wooster has obtained an interactive graphics terminal for use by faculty and students. The low-cost, direct-view storage tube terminal provides them with the capability to generate graphical results from statistical analysis, etc.

As discussed in a previous section, efforts are currently underway to implement a CAI system, PLANIT. An Otterbein professor is using his sabbatical to experiment with the system and develop several courses which he will share with the other schools.

The schools are also interested in data base concepts for their administrative applications. Although current plans do not include any major conversions in the near future, they are utilizing selected tools such as query languages and report generators.

SUMMARY

There are a number of other aspects of the program which deserve mention. One is that the availability of a large computer has been an important factor in faculty recruitment. For instance, one of the colleges was able to recruit an excellent young faculty member from a large university partly because they could provide him good turn-around on a large computer. The computer facility is also important in the recruitment of students, many of whom recognize the vast future importance of computer technology.

Faculty research is greatly enhanced. Members of the Wooster faculty have studied the life of Jesus by using the computer to analyze the words and phrases He used. Antioch faculty have developed simulations, particularly in the area of ecology. These programs, and many others, would not have been possible without a computer--and would have been difficult, at best, on a small computer.

Both simple and sophisticated administrative systems have been developed depending upon each school's needs. Written in COBOL, these are user-oriented systems that are designed for easy maintenance and modification. Administrative users do not have to memorize a lot of numeric codes "because the computer has trouble handling anything but numbers". Sizes of records and sizes of fields are not limited. The computer can determine "who are John's parents" or "does Bill Jones' employer have a matching gift policy", or "who might donate to the new theatre". Class lists are available prior to the first class. Acadmically deficient students are identified very early and are counseled. And most important of all, the computer does not make administrative reports "late".

Battelle is proud of BECUN and we believe that the colleges are also. The members are maturing rapidly in data processing. On the administrative side, they know more about themselves than ever before and academically they are teaching the use of the computer as a tool--as opposed to teaching programming. Faculty members are gaining increased recognition based upon research they have conducted with the aid of the computer. The program we believe is a great success and an excellent example of what cooperation among privately supported organizations can do.

A DATA BASE IS NOT A SYSTEM

Wallace C. Knapp
Director
Systems Development
The American University

Robert G. Ogilvie
Director
Computer Sciences
The American University

A DATA BASE IS NOT A SYSTEM

The American University purchased DOS TOTAL in September 1970, but did not implement a major data base until September 1973. This presentation reviewed the reasons for enchantment with data base management software in 1970 and the reasons for selecting TOTAL. The results of a major project in student records was evaluated in the context of the 1970 objectives. Unexpected problems, both technical and organizational, will be discussed. A new systems manager discussed his reactions to data base after assuming responsibility on the eve of implementation.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the authors.

MARK IV IN THE DATA BASE ENVIRONMENT

Tony Lamia
Director of MARK IV Applications
Informatics, Incorporated
Canoga Park, California

MARK IV IN THE DATA BASE ENVIRONMENT

This presentation explored the concepts of the data base as they relate to a file management system. The speaker illustrated how the MARK IV File Management System interfaced with data bases. Particular emphasis was given to IMS and TOTAL.

Copies of the slides used in the presentation are available upon request from the CAUSE Office.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

ARE WE TEACHING COMPUTER SCIENCE OR COMPUTER SCIENTISTS?

Alan Taylor
Columnist
COMPUTERWORLD

ARE WE TEACHING COMPUTER SCIENCE OR COMPUTER SCIENTISTS?

College courses for data processing instruct the student in computer science--but the use of computers takes place, not in a scientific vacuum, but in a real world situation which is very different from the environment assumed during the courses. When the real world environment, with its various commercial and social pressures, is fused to the courses without the differences being made explicit, a student can be lead to believe in a make-believe world. The speaker raised the question as to whether the courses are doing more harm than good in ignoring the real-world situation.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

EXECUTIVE PERSPECTIVES ON MIS

John L. Green, Jr.
Vice President
Administration and Budget
Rensselaer Polytechnic Institute

Henry G. Vaughan
Director
Management Systems & Analysis
Cornell University

Thomas S. Smith
President
Lawrence University

Herbert Zeitlin
President
Triton College

EXECUTIVE PERSPECTIVES ON MIS

John L. Green, Jr.
Vice President
Administration and Budget
Rensselaer Polytechnic Institute

Thomas S. Smith
President
Lawrence University

Henry G. Vaughan
Director
Management Systems & Analysis
Cornell University

Herbert Zeitlin
President
Triton College

A panel of experienced educational executives discussed their views on the relevance of the "informations systems" concept to the needs of executive managers of institutions of higher education. Topics discussed included a general description of key executive information needs, suggested organizational structures and responsibilities, and the general criteria by which college executives judge the effectiveness and efficiency of their information systems and administrative data processing staff.

Participants in the executive seminar were invited to prepare a copy of their main discussion points to share with readers of the *Proceedings*, however, not all statements were received in time for publication. The papers prepared by John L. Green, Jr., "Management Information Systems" and Herbert Zeitlin, "Key Executive Informational Needs for the Future" have been included.

Management Information Systems

Today's management, at almost all levels, finds it essential to have at its fingertips information with which they can manage the affairs of their organization. This need arises out of management's role in an organization which is to plan, to make operational decisions, and to evaluate performance. All three of these management functions require the manager to have access to information. Otherwise management would have to resort to making decisions arbitrarily, unilaterally, and intuitively which renders management inefficient and ineffective among other things.

When I speak of information for management, I am speaking of information in the broad sense of the term, namely data, facts, and information about developments and conditions in an institution, in other institutions, in the community, and in society at large.

Every organization finds itself having some form of management information even if management is unaware of its existence under this label. Ideally, the information is obtained as one of the outputs from a computer. However, this ideal situation does not exist in many organizations, especially not in small institutions where computer services are often provided by outside firms or the information is processed manually by members of the institution's staff.

In managing organizations, what is relevant to one manager may not be relevant to another. MIS, however, is relevant to the organization and it should be relevant to all managers. It is relevant to the organization in the sense that no organization can effectively function in today's society without information. Organizing the information for management's use has even more relevance in that it is essential to management in carrying out those three primary and essential functions of Planning, Making Operational Decisions, and Performance Evaluation. In my opinion, management seldom had too much information at its fingertips. The key is having it in a form which can be used and not in a form which requires intensive analysis. In other words, the analysis should take place prior to the time information reaches management.

To formalize the gathering and synthesizing of information for management, requires the assistance and the support of top-level management. This I assume we all know. The recognition by top-level management that it has an information problem is another matter. Where is the list of criteria that management should use to determine that an information problem exists? I have not seen such a list, have you? Perhaps the reason such a list is scarce, or maybe even non-existent, is because many administrators in higher education have not yet recognized nor accepted the fact that higher education needs an information system of some type for them to manage effectively.

Once management has decided to support the organized effort of developing an information system, an important first step is to determine structurally how the responsibility lines are to be established. In my opinion, there should be a single person charged with the overall responsibility of developing and administering the MIS. To whom this person reports depends on the institution, but it should be at a very high level. The plan for developing an MIS should include the following steps:

1) Formulation of Objectives
2) Development of a Data Dictionary
3) Determination of Information Requirements
4) Design of the Data Base System
5) Program Development and Testing

The objectives should be formulated by keeping in mind who the users of MIS will be. The data dictionary can be developed by using NCHEMS definitions where appropriate. Development of the data dictionary will assist management of the institution in easily identifying the information they want and in requesting same. I will not dwell any longer on the cook-book approach to MIS. Most of you know far more about this item than I do. I would, however, like to mention the methodology or procedure which I feel should be following.

Developing a computerized MIS is costly both from the standpoint of hardware costs and software costs. I am including in the software costs, the analytical work that is needed to prepare information for management's needs. We in higher education must find ways to decrease costs rather than increase them. So what can be done? I question whether a single institution can any longer justify the full costs of an MIS. Collectively, higher education can justify the most sophisticated MIS ever developed.

The bigger picture, therefore, is gathering information on a national scale because management is concerned with conditions and developments involving student disruptions, student newspapers, student fees, faculty salaries, unionization, institutional expenditures and revenues and the list goes on almost endlessly. What this implies is the need for a Higher Education Management Information System (HEMIS) which is available for all of management in higher education.

To accomplish this monumental task, we need information centers in each of the fifty states to collect and disseminate information. Each institution would pay a prorata share of the costs of the system. The Federal Government would also pay part of the costs since they have HEGIS and many other forms of information to collect and review. This is the methodology I commend to you for your consideration.

In Summary -

I think we all agree that MIS is relevant to management in every institution of higher education. The problem here may be that of educating management in the use of MIS.

What perhaps we are not in agreement on, is the state-wide and nation-wide approach to the development of MIS. My recommendation is heavily influenced by cost effectiveness. I simply do not feel that institutions can any longer justify the costs of having one or more computers on their campus. I would hope that institutions would cooperate in a venture to establish state-wide centers and a national network. Larger institutions would benefit by the cost savings and smaller institutions would benefit by having access to a computer at a relatively low cost.

Thank you.

TRITON COLLEGE
FACULTY BULLETIN-INFORMATION
Volume IX, Number 50
December 18, 1973

Subject:

KEY EXECUTIVE INFORMATIONAL NEEDS FOR THE FUTURE

Address Delivered by Dr. Herbert Zeitlin, President, Triton College
to the
College and University Systems Exchange
New Orleans, December 10, 1973

I come to you today not as one who is pessimistic about higher education but one who is optimistic. And as I look out into this audience I realize that there is much brainpower that will collect data to help us move more rapidly in the direction of universal higher education. Today, as you know, 10 per cent of the population over 25 have 4 years of college. In 2000 it is projected that 22 per cent of those over 25 will have 4 years of higher education.

My feelings toward the future of higher education are positive.

Whenever one is invited to address a conference of specialists and he doesn't know too much about the group, he'll always wonder, "Why was I asked?" In my particular case I'm not a specialist in management information systems or am I connected to any university as a research associate. I am president of an Illinois community college that has grown most rapidly.

In my state as in other states, community college faculty tend to feel sometimes that they are second-class citizens when compared to those employed by the universities. Our community colleges serve more than one-half of the students in public higher education and yet receive only 13 per cent of the total higher education budget. In years past the per cent was even less. Fortunately, today across the country community college education is "in" with the enrollments continuing to climb. State aid to the community college may double within the next decade. At least this one individual is going to try to make it happen!

Although we have heard much about the "Golden Years of Higher Education in the '60's" and the "New Depression in Higher Education for the '70's" as described by Earl Cheit, it is well to keep in mind that these general statements do not necessarily apply to all institutions.

Triton College in River Grove, Illinois, happens to be one of those institutions that is still in a golden period. We haven't reduced the budget or the staff. Perhaps a special thanks should be given to the staff in the office of information systems: they made it happen! We have grown from 1,200 students to over 17,000 within a period of 8 years. The assessed valuation has rapidly climbed from $800 million to $2 billion, while the operating tax rate has slowly moved from 15¢ to 20¢. Several communities have voted overwhelmingly to annex to us. Thus far we have built a $27 million campus. The master plan calls for $36 million. We are still in the building stages and expect 29,000 students by 1981. Tuition is now $9.00 per semester hour and will probably increase slightly by increments in the future.

We are labeled the Career Center of the Midwest by the Chicago Sun-Times because all of our degree and certificate programs are directed toward a career. Presently there are 140 such programs. Two weeks ago in its December 3rd issue, Time magazine featured Triton College in an article entitled "Shelves of Learning." Time compared Triton to a supermarket in that people in the community have everything they never thought a college could be: "inexpensive, accessable, and extraordinarily responsive [illegible] needs."

Last month while attending a conference of the Council of North Central in Phoenix I heard Dr. Louis Mayhew, Professor of Education at Stanford, elaborate on some of his findings. He mentioned that for some institutions that have not planned properly the new depression is very real. However, this fall he collected data to show that some universities are on the move again and the pessimistic attitude is declining.

In the '60's there was a high demand and a low supply of faculty, and students were standing in line waiting to get into the universities. In the fall of 1970, the long registration lines suddenly shortened and enrollments surprisingly dropped. Last year and the year before, university staffs actually turned to community colleges to increase their junior year enrollments. This fall the figures show that several universities are growing again and are optimistic about the future. Universities and 4-year colleges are making changes in traditional educational patterns to draw more students. For example: Dr. Mayhew pointed out vocational training programs heretofore reserved for 2-year colleges are now being extended to the universities.

Some of the advanced thinkers at Stanford are now exploring the possibility of a baccalaureate degree in 3 years. After all, if the better students now go to community colleges for a 2-year associate degree, why can't the universities be in a better position to attract this type of student by granting him a baccalaureate in 3 years.

Dr. Mayhew also feels that the concept of public universities and public community colleges satisfying all needs must go. Such a goal is too expensive! Directors of management information systems have known this for years. It is the responsible president who listens to his director as he points out the high-cost programs that are not meeting the needs of the graduates. I feel that it is morally wrong and a sign of fiscal irresponsibility if a president allows a program to continue, with the same budget allocation, when graduates can't be placed. Let me give you an example. This morning I heard that one of the great universities has decided to cut in half the admission of graduate students in Physics. Good! However, the size of the faculty and budget will remain the same.

I believe that state coordinating agencies of higher education must step in, with the data supplied to them by you, to prevent the duplication of expensive programs. There is no reason for several colleges within an hour's drive of each other to invest, for example, in very expensive numerical control, photo offset, or health services equipment so that each college could have a separate program for a few dozen students. The state must assert itself and authorize only a few institutions to offer the highly expensive lab programs. State officials should learn to say "NO" more often, particularly when they have valid data to substantiate a "NO" response. Perhaps your most important function is to supply state officials and college presidents with data so that the tax dollar is used wisely.

Now, what are the key executive informational needs for the future?

As many of you know, universal access to higher education has been a central concern of the Carnegie Commission on Higher Education when it was first organized in '66. Last month the Commission released a new publication entitled Priorities for Action: Final Report of the Carnegie Commission on Higher Education. This report, I believe, should be read thoroughly by anyone engaged in management information systems because it discusses future possibilities in higher education. Without question, before decisions can be made by state officials, college presidents, boards, and faculties, valid data must be obtained -- obtained rather quickly, interpreted, and acted upon. On the immediate horizon your office may be requested to obtain data on any one of the following concerns of the Carnegie Commission. Here are 9 priorities for research:

1. Need for Health Career Personnel

   There is presently a shortage of health career personnel. Higher education now has the major responsibility for the training of such personnel. Money will be made available through state and federal agencies to help you find out what is needed in your community.

2. The Changing Job Market

Every college must make a serious study of the changing job market as it reflects the needs in your region. Job opportunities for college graduates began turning downward in 1968 and a lack of active demand has continued.

3. Collective Bargaining

Unionization of college faculties has started. Collective bargaining could have a major impact on higher education as it did earlier in American industry.

4. Women's Liberation Movement

Women are demanding more access to graduate schools and to faculty positions. The universities of old have been mainly male preserves. In 1970 approximately 22 per cent of college and university full-time staff were women. According to the Carnegie projection this figure will increase to 28 per cent in 1980, 29½ per cent in 1990, and maybe 39 per cent in the year 2000. To some this projection is a sign of male chauvinism. Thirty years to reach 39 per cent! I know a segment of the women's liberation group in my area that feels 39 per cent is just 20 years too late. Thirty-nine per cent should be in 1980 and not in the year 2000, and it may well be!

5. Absorption of Minorities into the Mainstream

Presently minorities make up approximately 15 per cent of our population. College and university faculties now employ about 5 per cent. Where are the minority groups at this convention? The projection for the year 2000 is over 15 per cent. Again, this projection by several minority groups is not acceptable. If the year 2000 projection could be made as a goal for 1980, the minority unrest may be alleviated.

6. Rising Interest in the Welfare of the Total Post Secondary Age Group

There will be continuing interest in the welfare of the post secondary age group. Higher education will not be reserved only for the recent high school graduate.

7. The Role of State and Federal Influence Towards Higher Education

The great society is giving way to the new federalism. Attention shifts from what the federal government might do for higher education back to what the states should do. And today there is the trend towards state coordinating agencies to master plan all segments of higher education.

8. New Enrollment Patterns

The Carnegie Commission was the first to call attention several years ago to the '80's as a "stop" period after a century when full-time enrollments doubled every 10 to 15 years. The Commission, even with its vast resources, did not anticipate that the declining enrollments would occur so soon and so fast in the '70's. Consequently, the Commission now cautions against planning for new campuses. However, stand by for the great invasion of the part-time student who attends 1/2 day, evenings, or weekends. The more mature student, something referred to as the post secondary group, will be demanding college credit for non-traditional experiences.

9. New Mentalities: Climbers and Sliders

A new concern of college campuses is the attitude of the "sliders." These are students who accept or even insist upon a lower socio-economic status than their parents. "Sliding" can be a painful experience for individuals involved. And some campuses may need to adjust to the special problems of such students; they have been more oriented towards the "climbers."

These are nine areas which the Carnegie Commission has mentioned as future possibilities. Without question before college presidents can act intelligently on any of these possible trends more detailed studies have to be made. The relationship between the director of management information systems and college presidents will have to be a clo se one. But I am confident that with the vast tools at our disposal from IBM and other companies, meaningful data can be obtained and interpreted properly so that the right decisions will be made.

H7/mag

Distribution: Bd, JtC

PLANNING AND CONTROL OF DEVELOPMENT

Herbert W. Bomzer
Assistant Director
Administrative Data Processing
University of Illinois

PLANNING AND CONTROL OF DEVELOPMENT

The Management Communications System (MCS) designed at the University of Illinois facilitates planning and control of the administrative systems development process. This system is designed to encourage planning and exchange of information by development personnel at all levels. The system is based on experience with Gant chart, PERT, and critical path techniques.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

A DATA SET MANAGEMENT SYSTEM

E. M. Staman
Computer Center
Union County Technical
Institute

B. E. Squires
Computation Center
The Pennsylvania State
University

A Data Set Management System

Introduction

A system is described for the automatic management of a large number of data sets. It is generally applicable to any situation in which the generation and retirement of data sets is predictable, and in which certain computer programs or procedures are executed on a more or less scheduled basis. The system includes provisions for unscheduled data set identification, retrieval, and verification.

The term management, when applied to data sets, is meant to cause the proper and automatic selection of data sets for input and output of computer programs, appropriate generation of back-up copies, and proper removal of the data set from the system when the necessary requirements for this removal have been met.

This system has been implemented as part of a billing, accounting, and management information system involving 1.5 million charges annually. It is in use at the Computation Center of The Pennsylvania State University and embraces about 200 reels of magnetic tape data sets. Because the effectiveness of any system is measurably affected by its implementation, brief discussions on the implementation of this system are included within this paper.

Problem to be Solved

The system described in this document involves a minimum of eight different types of data sets comprising a minimum total of at least two hundred data sets with an approximate increase of twenty data sets per year. Depending upon its information content, a data set may be updated, copied, or saved for a period of time extending from one day to seven years.

This situtation, probably typical of applications involving large numbers of data sets, generates several problems. For example, the problem with constantly changing JCL (it inevitably gets changed incorrectly frequently), the problem with trying to manage 200 tapes by following their uses in a notebook -- the problem of keeping the notebook up-to-date. The problems associated with several persons accessing these tapes while they are changing and one person not being 100% certain what everyone else is doing -- perhaps not getting the data he was really after. The problem of not being quite certain just when a tape was available for reuse and the problem when a tape which is not really available gets reused anyway. In general the problem at hand is to design and implement a subsystem to a Data Processing System to aid in the automatic management of data sets used in the system.

Analysis and Design

An analysis of a data set management problem will raise a least four distinct questions.

1. What decisions must be made concerning the data sets in the system?
2. What information must be present in order to be able to make these decisions?
3. How can this information be effectively communicated among the programs in the system?
4. What are the present and potential relationships among the data sets in the system?

The desired end product of a system to automatically manage data sets can be summarized as follows:

A program must, at its own discretion be able to select any data set that is required for its successful termination. Furthermore, provision must be made for the removal of redundant data sets from the system, and there must be sufficient checks and balances within the system to prevent erroneous selection or removal of data sets. The checks and balances should include a technique for automatically determining the currency of the data and the "format level" of the data, as well as the usual counters and date indicators.

The decisions that must be made, therefore, are precisely the same set of decisions that a programmer or operator must make when he selects data sets to be used in conjunction with the execution of a program. Theoretically he askes himself:

Which data is to be input? Do I have all of it? Where can I put the output (for example which tape is available for output?) Has the format of the data changed so that this program will not run without modifications?

In practice, once the system becomes complex enough, the programmer or operator may ask the wrong questions or surmise the wrong answers, resulting in unnecessary reruns of programs, incorrect output, or in the worst case, destruction of data resulting in the initialization of usually costly backup procedures.

The general answer to question #2 is: Those parameters required to insure that a data set contains the correct control information with respect to the quantity of information being processed and the time frame of the information. Also required is the correct information pertaining to currency of data sets, version of format (has the format of the data set been changed?), and availability for disposal. In the particular system discussed in this paper a record with eleven fields was created which indicated the status of each data set in the system.

The information contained within the records follows:

1. TYPE: A code indicating which type of data set is represented in this record. (i.e. master, accounting, data base, etc.)

2. BEGINNING DATE: The date marking the beginning of the period that the data on the data set was produced.
3. ENDING DATE: If the data set is rotated or updated then the ending date is the last date that the data set was updated. Otherwise, the ending date is the date marking the ending of the period that the data on the data set encompasses.
4. NUMBER OF CHARGES: For accounting purposes. This represents the number of charges represented on the data set.
5. NUMBER OF RECORDS: For control purposes. This is the number of logical records on the data set. Thus one can always determine that the proper number of records have been processed at the end of a production run.
6. IDENTIFIER: An identifier for the data set. This identifier is unique with respect to production programs; that is - no program in the system will ever require two data sets with the same identifier.
7. AVAILABLE: A code indicating-
    a) the data set is available for reading only.
    b) the data set is available for read or write.
    c) the data set was being used when there was an abnormal termination, system crash, etc. Interrogation of this code makes restart possible.

8. IDENTIFIER OF DUPLICATE: The existence of a duplicate copy of the data would be recorded in this field.

9. AVAILABILITY OF DUPLICATE: Like "AVAILABLE" - for the duplicate copy.

10. LEVEL: =K. The K=th version of the format of the data set since the beginning of the system. Interrogation of this code tells the program how to read the data.

11. CURRENCY: =on if all data on this data set is current.
    =off if data is detected elsewhere in the system which would affect this data set.

Thus one can see that the recorded information for a data set contains enough information to allow the necessary control for the data set to be carried out effectively, as well as the information required to permit the necessary management decisions pertinent to the data set.

For example, suppose a quarterly data base is to be created from three monthly data bases. If the quarterly data base is sucessfully created, the program creating the tape could update the parameters by releasing the corresponding monthly data base tapes, thereby making them available for future use. This can be done by calculating the dates associated with the data bases to be released and changing the availability indicator on the record associated with those data sets.

An important note at this point in the analysis is that the system being developed must retain the flexability to enable the manager to superimpose restrictions (for control purposes for example) upon the management of the data sets.

Thus in the above example involving monthly data base tapes it was decided that the criteria for the release of the tapes for use by other programs should be that a duplicate copy of the quarterly tape (created from the monthly tapes) has been sucessfully read by a program. The execution of the criteria must be done automatically.

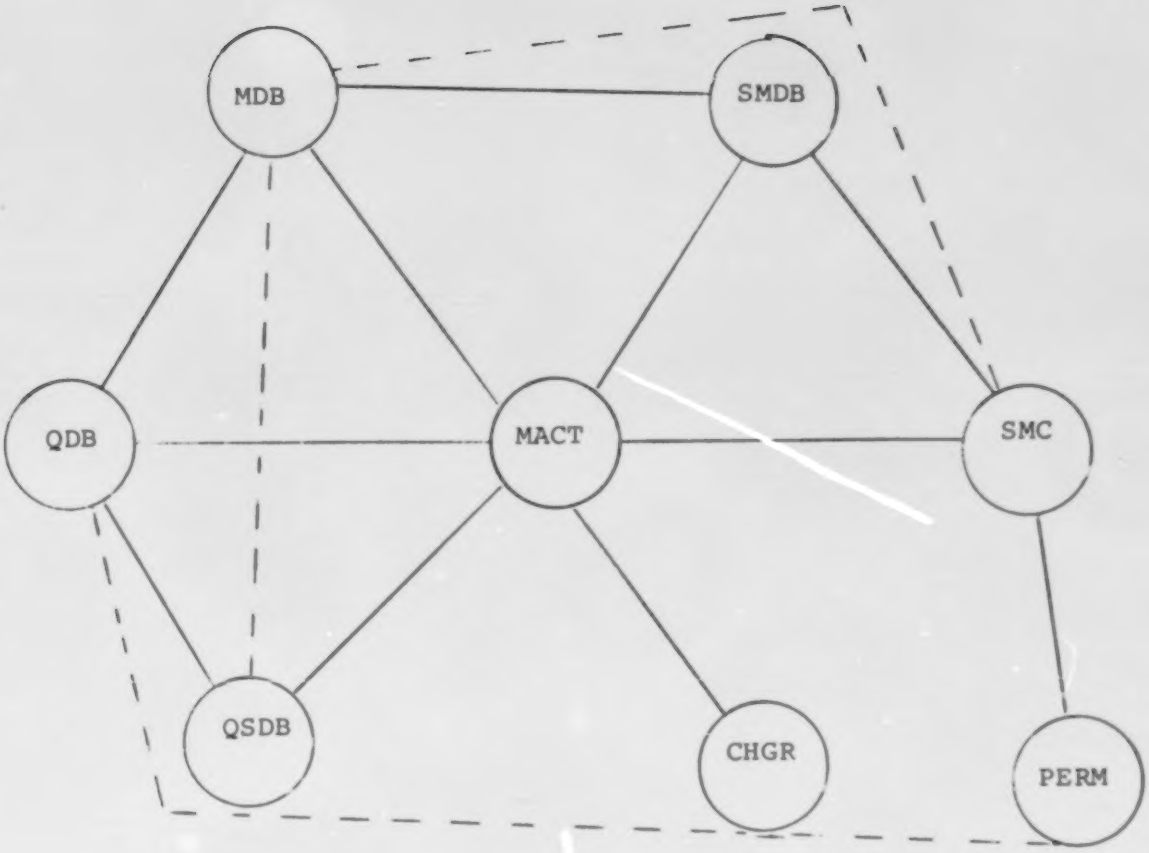
In an attempt to discover the answer to question #4, the following steps were taken:

a) Classify the types of data sets in the system with respect to content of data. (master accounting, charges, data base, etc.)

b) Classify the life of the data in the system. (this includes both the useful life in terms of report generation as well as the required life of certain data because of legal restrictions.)

c) Finally determine a scheme to uniquely identify any subgroup of the group of data sets in the system.

Figure #1 is a diagram generated during the analysis required for question #4. The development of this diagram is based upon ascertaining which data sets are needed or will ever be needed simultaneously by any job in the system. When the relationships among the data sets in the system became known, ranges of unique identifiers were assigned to classes of the data sets.



KEY:

———	Data sets are presently required simultaneously by some given job in the system
-------	Data sets may in the future be required simultaneously

MDB	Monthly data base	SMDB	Semi-monthly data base
QDB	Quarterly data base	MACT	Master Accounting tape
CHGR	Charge records	PERM	Permanent records
SMC	Semi-monthly charges	QSDB	Quarterly summary tape

Present and Potential relationships among data sets in the Job Accounting System.

Figure 1

Implementation

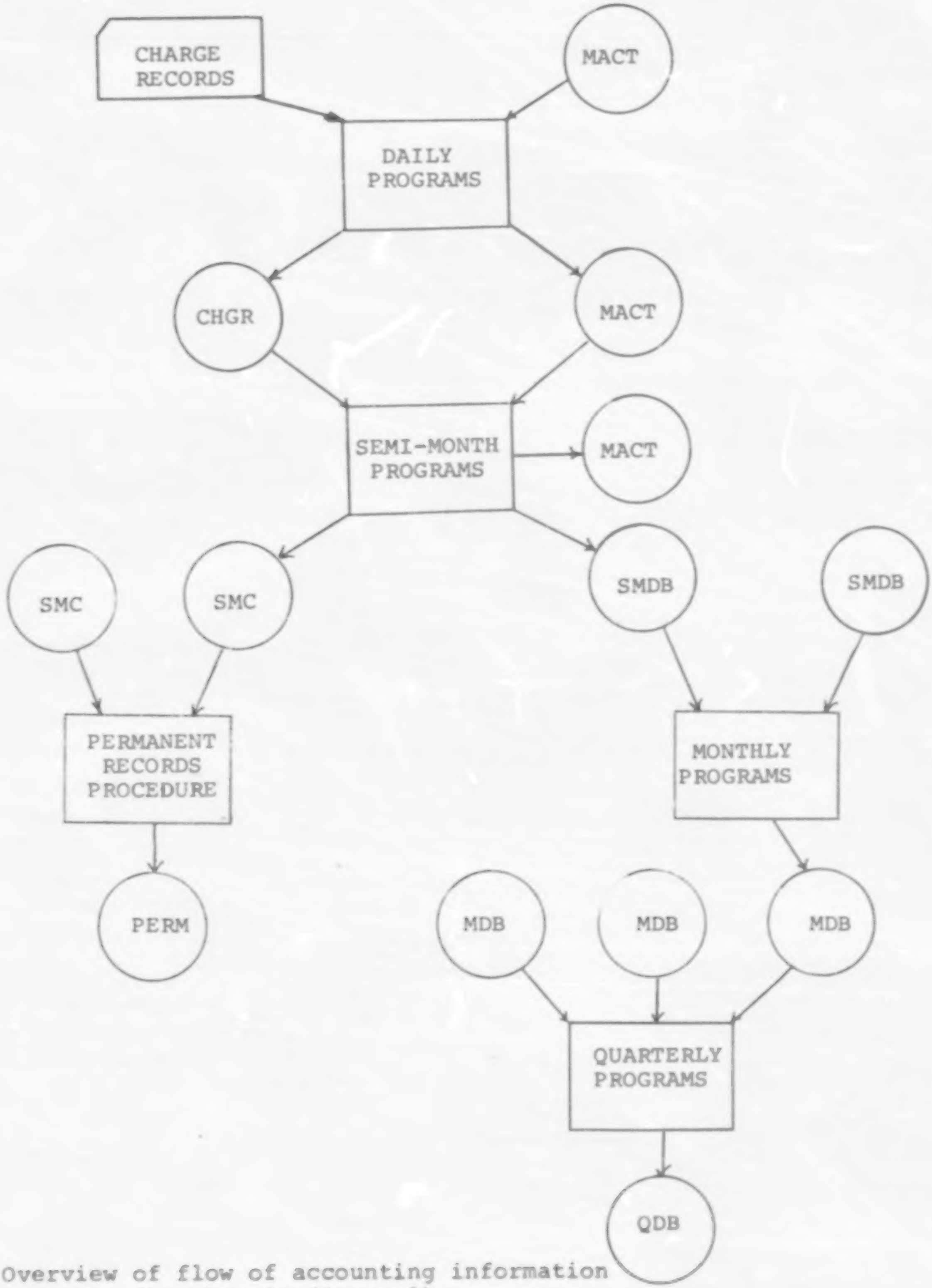
Typically maintenance of job accounting information begins with the creation of a charge record and ends with permanent tape records of the original information. In addition, the normal course of processing would probably reduce the data to some sort of data base form, preferably as soon as possible within the system. A simplified overview of the system for which the data set subsystem was designed is contained within the following narrative and in figure 2. The narrative is broken into four time frames, each a major breakpoint in the system.

Time frame 1: Daily Programs.

The original charges for the system are on charge records which are concatenated to tape on a daily basis. One of the master accounting tapes is also updated at this time, and rejected charges are removed from the system for subsequent hand processing. The expected life of a master accounting tape is about four days.

Time frame 2: Semi-monthly programs.

A summarized version of the semi-monthly information in the form of data base information is produced. The data base contains only 15% as many records as the original tapes, with each record reduced approximately 30% in length. Thus there is about a 90% reduction in the quantity of information to be carried to report producing progams from this point.


CHARGE RECORDS
MACT
DAILY PROGRAMS
CHGR
MACT
SEMI-MONTH PROGRAMS
MACT
SMC
SMC
SMDB
SMDB
PERMANENT RECORDS PROCEDURE
MONTHLY PROGRAMS
PERM
MDB
MDB
MDB
QUARTERLY PROGRAMS
QDB

Overview of flow of accounting information
(see key in figure 1)

Figure 2

A backup of the master accounting tape is created at this point, with an expected life of four weeks.

Time frame 3: Monthly programs.

The semi-monthly data bases are concatenated into a monthly data base and discarded at this point. Also the semi-monthly charge tapes are concatenated onto permanent record tapes to be saved for seven years. Monthly billings are produced during this phase of the system.

Time frame 4: Quarterly programs.

The monthly data bases are concatenated onto a quarterly data base and discarded at this point. A quarterly summary data base is produced.

The problem of implementation raises the following questions:

1. What type of device shall be used to carry the data?
2. How shall such a device be organized?
3. What language shall be used to access the information?
4. Shall the code be inline, subroutine, etc.?

Certain items are obvious. In particular, a direct access device is surely the place to put the data. The records involved are both short in length and few in number. The question of data set organization, however, is not clear. Even though the data set is small and is accessed relatively few times, some interesting questions are raised when one considers the merits of direct access vs sequential access.

Perhaps the first question to be considered is the one of backup. The Job Accounting system was already designed so that a decision to save the entire contents of the data set on a semi-monthly basis could be easily implemented no matter what the final organization becomes. However, since the data set will be updated on a daily basis, two cases should be considered.

1. A sequential file must be read entirely and should be recopied to another file every time it is updated. Furthermore, since it is not known until the entire data set is read just which record must be updated, the entire data set would have to be in core at once, prior to rewriting. However, the obvious advantages in speed and backup of a sequential file do exist.
2. A direct access file can be updated in place, so the core required is for only one record at a time plus one or two pointers to help remember which record is finally to be updated. Loss of the file, however, could be catastrophic. Such a loss might require retreating to the last time the file was saved for back-up purposes. However, upon further considerations, it was decided that the entire file (less than 200 records) could be printed by every job that ever updated the data set, and that backup would involve keypunching no more than 200 cards

and recreating the data set.

The decision was made to go with direct access in lieu of sequential access. (Incidentally, the data set has not yet been lost so backup has not yet been necessary).

Fortran was chosen for the language for several reasons, the most overwhelming of which was that 95% of the programs requiring the data are also in Fortran. Furthermore, implementation, maintenance, and documentation are easier in Fortran than Assembler (the only other reasonable choice.) Since a multitude of programs need to access the data, the decision to create a sequence of subroutines and place them on the library certainly seemed the best method of implementation.

By now it is probably apparent that one point of the system is to effectively utilize the unique identifier assigned to the data records by identifying the proper symbolic JCL with the numbers. Recall, in Fortran, a data set can be identified by:

```
//DATA.FTnnF001 dd ...
```

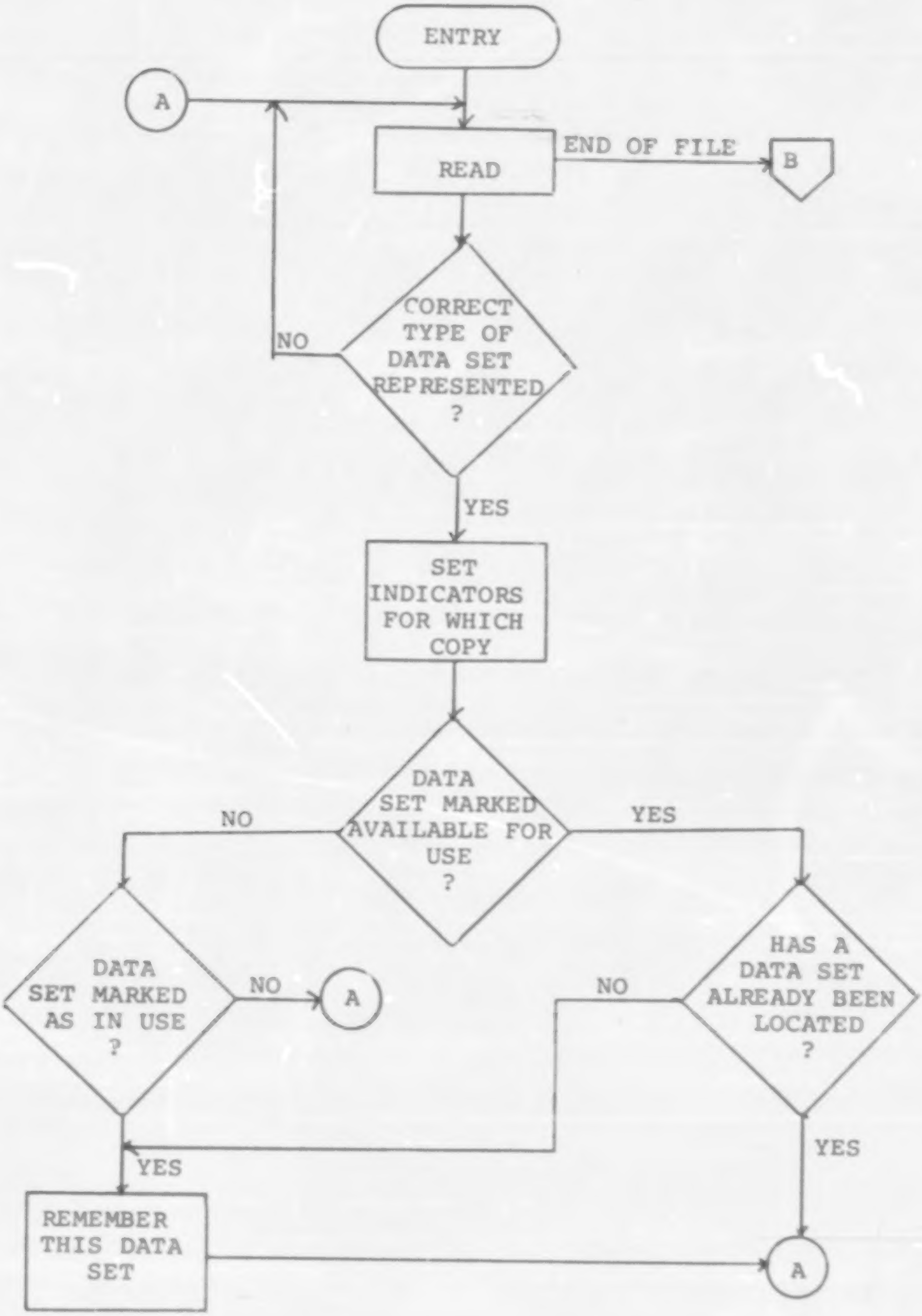
where nn is a two digit number referenced in the Fortran program for a read or write. The same type of symbolic JCL scheme can be applied to other processors.

A sequence of five subroutines was created for data set management. A brief description of these routines follows:

GETAPE: search the management data set for an available identifier to use for a data set to be created. Mark the availability bit of the number to indicate that it is being used. In essence, GETAPE must be responsible for returning an identifier of a tape available for either an original or a duplicate copy of a data set. If the program had been restarted for some reason, GETAPE must be able to return precisely the same identifiers that were being used during the previous execution of the program. Figure 3 is an overview of the logic of GETAPE.

FINDTP: search the management data set for a data set of a given type and date. This is -- find the identifier of a previously created, ready to process data set, and return the identifier to the main program. The rule for FINDTP is that it should always return an identifier for the duplicate copy of the data set, if such a copy exists.

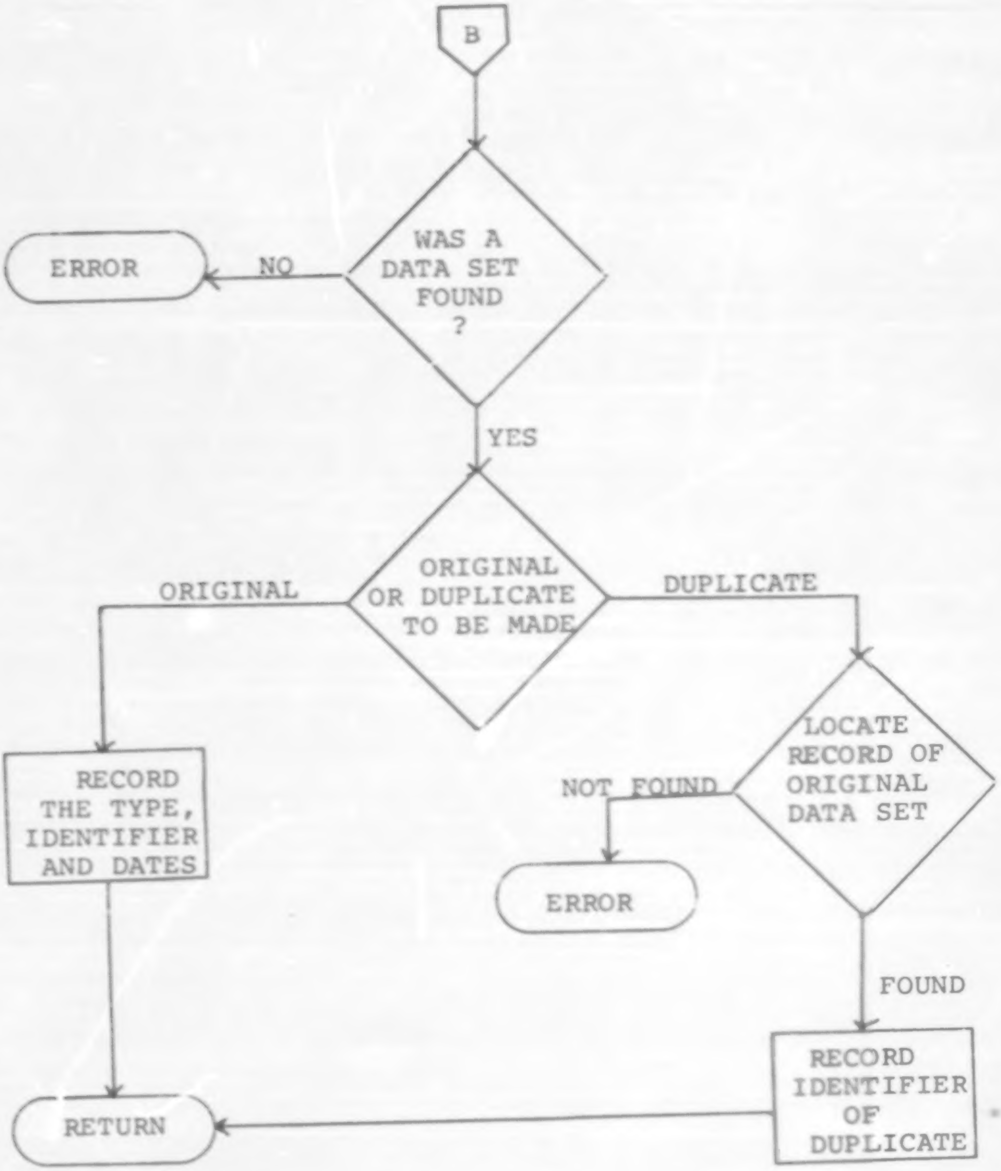
PUTAPE: update the data set with given information on a newly created data set. Include the given number of records, number of charges, etc. Mark the availability bit to say that this data set is a duplicate copy.


ENTRY
A
READ
END OF FILE
B
CORRECT TYPE OF DATA SET REPRESENTED ?
NO
YES
SET INDICATORS FOR WHICH COPY
DATA SET MARKED AVAILABLE FOR USE ?
NO
YES
DATA SET MARKED AS IN USE ?
NO
A
HAS A DATA SET ALREADY BEEN LOCATED ?
NO
YES
YES
REMEMBER THIS DATA SET
A

General GETAPE logic

Page 1 of 2

Figure 3


B
WAS A DATA SET FOUND ?
NO
ERROR
YES
ORIGINAL OR DUPLICATE TO BE MADE
ORIGINAL
DUPLICATE
RECORD THE TYPE, IDENTIFIER AND DATES
LOCATE RECORD OF ORIGINAL DATA SET
NOT FOUND
ERROR
FOUND
RECORD IDENTIFIER OF DUPLICATE
RETURN

General GETAPE logic

Page 2 of 2

Figure 3

Be sure that the control information is correct prior to recording the existence of the data set. Figure 4 is a generalized view of PUTAPE logic.

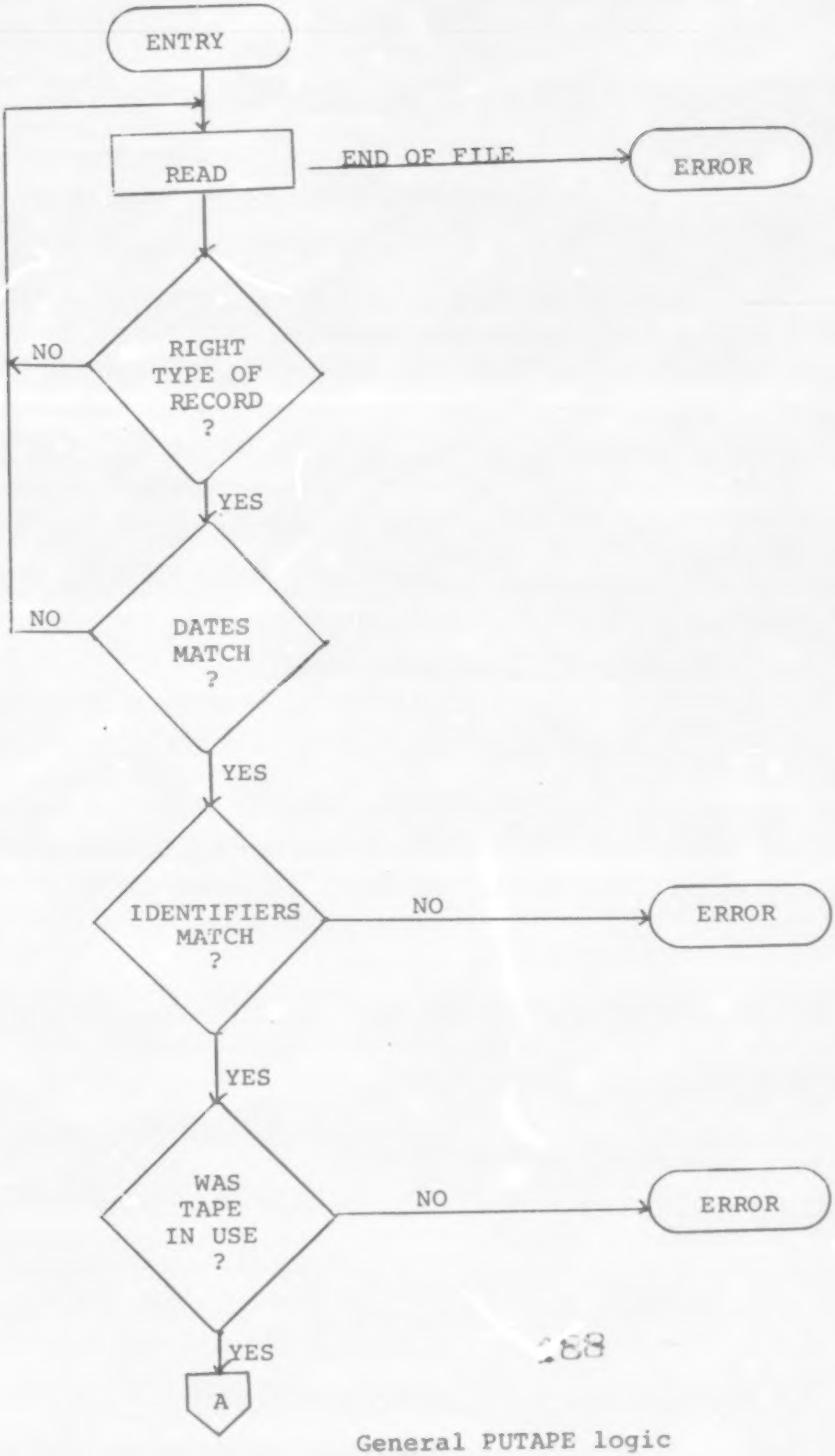
FREETP: release the data sets associated with the given number, type, and dates. Do this by marking the availability bit to say that this data set is available for use.

LISTP: output data set management information on a given device.

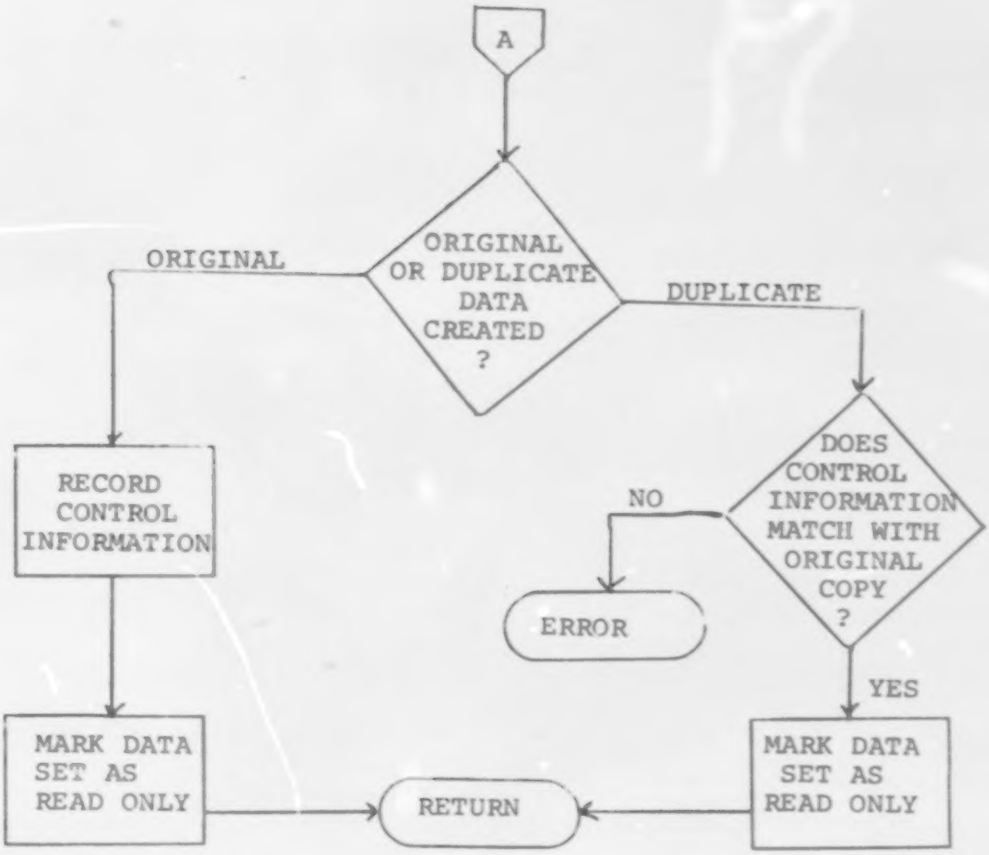
As would be expected, several "utility" programs to aid in the implementation were created during the initial stages. These are not considered part of the subsystem and will not be discussed in this document.

Conclusion

The requirement for a data set management system led to the construction of the subroutines described in this paper. One would suspect that these requirements are not unique to the application leading to their development. For example, would the routines not be just as valid for an organization doing rotating billing, say A - C on the first of the month, D - F on the 2nd, G - H on the 3rd... W - Z on the 30th?


ENTRY
READ
END OF FILE
ERROR
NO
RIGHT TYPE OF RECORD ?
YES
NO
DATES MATCH ?
YES
IDENTIFIERS MATCH ?
NO
ERROR
YES
WAS TAPE IN USE ?
NO
ERROR
YES
A

General PUTAPE logic
Page 1 of 2
Figure 4


A
ORIGINAL OR DUPLICATE DATA CREATED ?
ORIGINAL
DUPLICATE
RECORD CONTROL INFORMATION
DOES CONTROL INFORMATION MATCH WITH ORIGINAL COPY ?
NO
ERROR
YES
MARK DATA SET AS READ ONLY
RETURN
MARK DATA SET AS READ ONLY

General PUTAPE logic

Page 2 of 2

Figure 4

Figure 5 in the diagram demonstrates how one might in general use the subroutines. The purpose of the figure is to aid in the indication that conversion of existing programs to use the routines involves the following steps:

1. Placement of call statements at strategic locations within the program.
2. Modification of existing read and write statements, if necessary, to access devices represented by the symbolic identifiers returned by the subroutines.
3. Expansion of the JCL in the programs to include all the data sets that are represented by the appropriate identifiers.

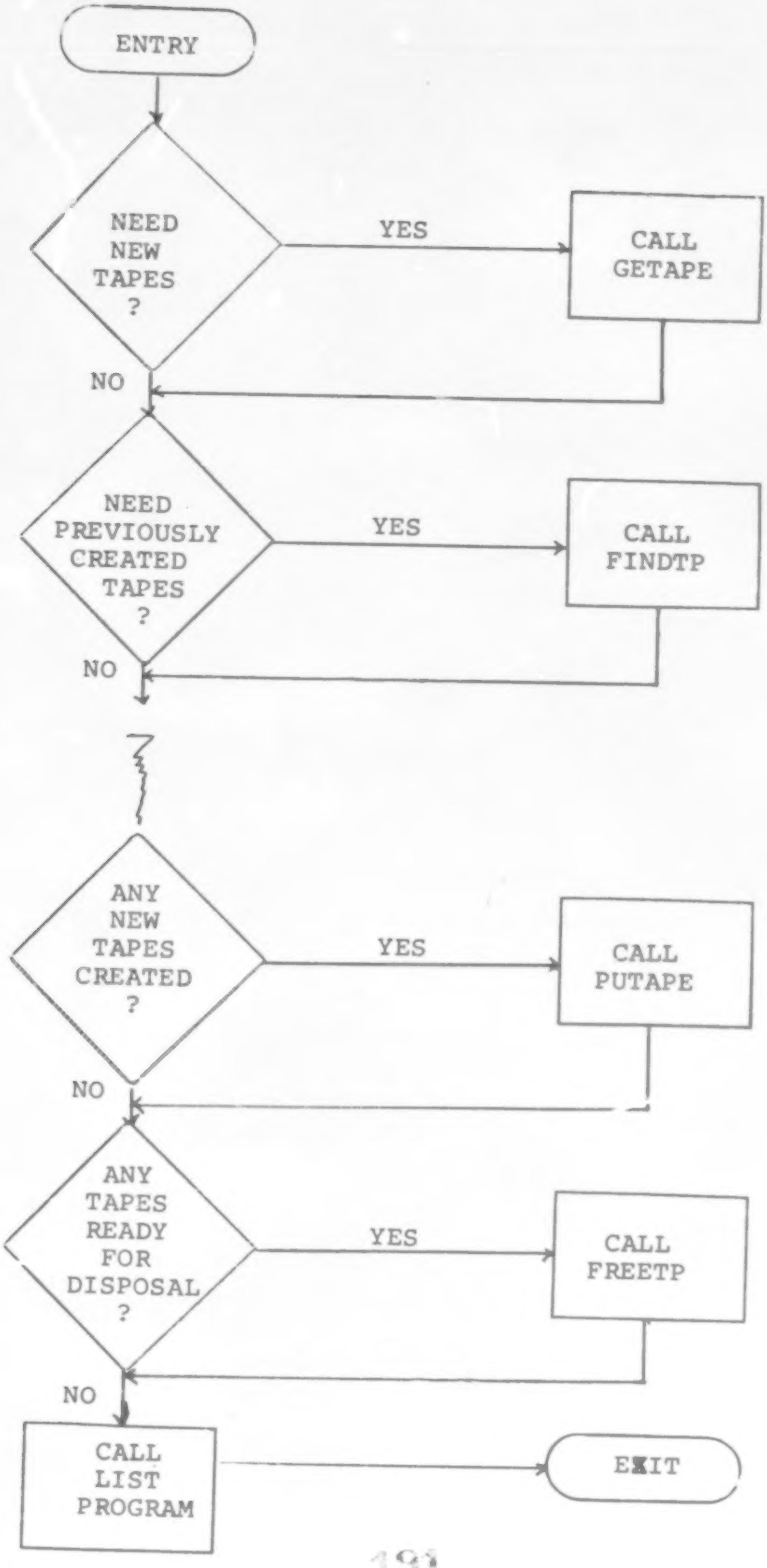

ENTRY
NEED NEW TAPES ?
YES
CALL GETAPE
NO
NEED PREVIOUSLY CREATED TAPES ?
YES
CALL FINDTP
NO
ANY NEW TAPES CREATED ?
YES
CALL PUTAPE
NO
ANY TAPES READY FOR DISPOSAL ?
YES
CALL FREETP
NO
CALL LIST PROGRAM
EXIT

Figure 5

DBMS EVALUATION PANEL

Kenneth E. Shostack
Application Development Group
Office for Information Technology
Harvard University

R. Brian Walsh
Director
Computing Center
University of Notre Dame

John L. Sullivan, Jr.
Director
Information Systems Center
University of Rochester

L. Carvel Whiting
Director
Systems
University of Utah

DBMS EVALUATION PANEL

Kenneth E. Shostack
Application Development Group
Office for Information Technology
Harvard University

John L. Sullivan, Jr.
Director
Information Systems Center
University of Rochester

R. Brian Walsh
Director
Computing Center
University of Notre Dame

L. Carvel Whiting
Director
Systems
University of Utah

There is a great deal of interest among colleges and universities with respect to the potential advantages of Data Base Management Systems. Many institutions are exploring the possibility of utilizing such a system in the administrative environment for improving information flow. The question most installations are faced with is how to evaluate and select the appropriate system for their particular institution. This panel, composed of four representatives of leading institutions that have done systematic studies of Data Base Management Systems reviewed the techniques utilized and data collected by their respective institutions in determining which system would come closest to satisfying the needs of their institution.

Participants were invited to prepare written summaries of their comments. Unfortunately not all statements were received in time for publication. Included here are the statements by Kenneth E. Shostack, "One University's View of Data Management Systems;" John J. Sullivan, "A Suggested Acquisition Procedure; and R. Brian Walsh, "Data Base The Concept The Commitment".

One University's View of Data Management Systems

by

Kenneth E. Shostack
Staff Analyst

Office for Information Technology
Harvard University

Introduction

Because of the current financial situation, many colleges and universities are re-examining their approach to administrative data processing. One concept most frequently offered for improving services is that of data management systems, sometimes called data base management systems. At Harvard University we were asked whether a data management system would alleviate the three problem areas of maintenance, timely reporting, and data entry. We were further asked which data management system would best contribute to the alleviation of those problems. In order to respond to these questions, we had to define our problems and study the available data management systems. This paper describes some of the problems being faced by Harvard and discusses those aspects of the evaluation of available data management systems which are not normally described in the literature.

At Harvard University we are facing three problems, which are being faced by most data processing departments. The first of these problems is that of application system maintenance. As shown by several recent studies, the costs associated with data processing have been shifting. In the early 1960's the most costly item in the data processing budget was the cost of computer resources. By the mid 1960's this had shifted; the most costly item in the budget was applications development. Today, the most costly item in the budget is that of maintenance which involves the adding, deleting or modifying of data elements in a file, i.e., changes which usually require modifying every program in the system. One recent study[1] claimed over 50% of the data processing budget in a mature operations organization is for maintenance. Another 30 to 40% of the budget is for operations. This leaves only 10 to 20% of the budget for developing new applications.

---

[1] "First Warned to Design Systems for Small, Unsophisticated User", Computerworld October 3, 1973 Pg. 27.

A second problem is that of preparing timely reports, especially new reports or one time reports. Historically, requests for reports were fulfilled by developing programs. Usually the need for the reports ceased to exist before the report was ready. More recently report writers have been available. Report writers enable programmers or analysts in user departments to prepare reports in a matter of hours or days rather than weeks. Users want more timely information, such as that available with on-line terminals and query languages.

A third problem is getting transaction data into the data files or data base in an efficient and timely manner. There are many parameters to measure whether a method of data entry is efficient, and there are many philosophies regarding approaches to data entry. Currently, all administrative data entry at Harvard is accomplished using keypunches or a key to disk system. The cost effectiveness of alternative data entry technique, including on-line data entry needs to be explored.

In order to assist in the discussion of the requirements for a data management system and the capabilities offered by various vendors, a framework for analysis was developed. This framework is based on a functional definition of the ideal or complete data management system. The complete data management system (Fig. 1) is a set of computer programs which provides access to an aggregate of data elements called a data base.[1] The data base management system provides a method of gathering data elements, eliminating duplication of information, reducing file space, and achieving greater accuracy (by reducing redundancy).

The data is normally stored on a disk in a random manner which provides for:

A. Sequential access for volume processing,

B. Random access based on multiple access points (inverted value tables) to facilitate random retrieval and maintenance.

Security of confidential information is maintained at the data element level as well as at the data base level.

Services are provided for generalized and tailored access to the data base. These services are available in on-line mode via a teleprocessing monitor from various typewriter and video terminals or in batch mode for high volumes of transactions. The generalized services include data base inquiry, data base update, and the quick specification and generation of reports through the user language interpreter. The tailored services are user written application programs, created to perform functions through procedural language interfaces.

---

[1] This description is partially extracted from "OASIS, Briefly Stated" in the OASIS Programmers Reference Booklet.

The central facility of a data base management system is the data access program which accepts requests from higher level programs and translates them into detailed manipulations of data elements in the data base. Requests can refer to data elements by name, without regard to their physical location within the data base. Thus, it is possible to deal with the data in a logical manner without regard to data base organization.

This framework can be used to discuss the four products which we examined in depth. (Figure 2).

Product	Vendor
IMS, GIS, IQF	IBM
MODEL 204, IFAM	Computer Corporation of America Cambridge, MA
System 2000	MRI Systems Corp. Austin, Texas
TOTAL, Environ/1	Cincom Systems, Inc. Cincinnati, Ohio

A fifth system, OASIS, which was developed by Stanford University, was also considered. OASIS is an excellent system, but it does not now offer the flexibility required to fit into Harvard's plans. Any college or university considering a data management system should give OASIS some consideration.

As a result of an examination of these systems, the following general observations were made:

1. No system offered all the features desired by Harvard University.
2. Except for operating systems, data management systems are the most complex software systems available. Few installations choose an operating system independent of the hardware.
3. The monthly rental for a complete package from the four vendors ranged between $3,000 and $3,400. The monthly rental for the features which Harvard would acquire ranged between $1,700 and $2,400.
4. The introduction of data management systems appear to have the following effect on applications systems costs:

A. Development Costs	
1. Procedural Language Applications	20% Lower
2. Special Language Applications	60% Lower
B. Operating Costs	20% Higher
C. Maintaining Costs	20 to 40% Lower.

Thus the costs are shifting from the labor intensive processes to machine intensive processes.

5. By eliminating data redundancy, the department which maintains the file would have the most efficient access to that file. Other departments would have less efficient access, but would be able to eliminate the clerical tasks required to maintain the data.

Recognizing the large differences in capabilities of these systems and the variety of techniques used to implement any given function, it was decided that the best method for obtaining an understanding of each system would be obtained by running some tests using data from existing systems. The tests were not intended to be complete tests of each system, nor were they intended to show all the strengths and weaknesses of each system. They were intended to provide a deeper understanding of how these systems could be used, and to augment the conversations with users, and to augment the study comparing of the features of each system. Rather than documenting a matrix of features and a description of user conversations, I would refer the reader to the documents being compiled by the CAUSE Data Management Systems Task Group which are being made available through CAUSE and the reports of the CODASYL Data Base Task Group which are available from the ACM.

Harvard University currently maintains two personnel record systems. One system is for salary and wage personnel. The second system is for employees who hold corporation appointments. The initial application schedule for implementation using a data management system is an Employee Information System, which will be an upgrading and integration of the separate personnel systems. Mrs. A. M. Koss, who is a Senior Staff Analyst in the Applications Development Group, was the project leader responsible for developing the wage and salary personnel system, and is the project leader on the new Employee Information System, designed the tests using data from the two existing systems. The tests included loading 500 records and performing 53 transactions against the data query and up-date capability.

The fifty-three transactions breakdown as follows:

Class	Quantity
insert	10
delete	7
replace	7
retrieve	29

The costs of loading the data are shown in Figure 3. The transaction performance for processing transactions is shown in Figure 4. For simplicity, the charts do not show all components of the cost, but rather the more common components of cost, cpu, and disk accesses. System 2000 can be configured to run in various size regions of core by use of program overlays. The test run at Harvard (Figures 3 and 4) were run in the smallest (136k) and largest regions (314k). The performance differences are the most pronounced on the tests of transaction performance (Figure 4).

One item of interest from the tests was the ability to develop applications programs in 60% less time with the special languages provided by System 2000 and Model 204, than with COBOL as required for IMS and TOTAL.

In addition to the tests described above, two additional tests were performed with System 2000 and Model 204/IFAM. These two tests (Figure 5) involved the fifty-three transactions with a batch COBOL and also with the user language under TSO in an interactive mode. In both these tests the highly overlaid version of System 2000 was used.

The Harvard Computing Center operates an IBM 370/145 which is connected via microwave link to an IBM 370/165 at MIT. The 165 has Time Sharing option (TSO) and the low speed telecommunication access ports available to users. An analysis of the costs for dedicating a region and low speed telecommunication posts on the 145 to data management system and the costs of operating a data management system under TSO, showed that until there were six terminals on-line three hours per day it could be more cost effective to operate in the TSO environment. The disadvantage would be that each user would have his own copy of the data management system software, so there would be a potential problem of two users attempting to update the same record. A further restriction of the TSO implementation is that users are normally restricted to regions of less than 200k. There are solutions to this potential problem. So it was decided that, initially, TSO would be the teleprocessing monitor.

As a result of this effort Harvard University determined that MODEL 204/IFAM would provide the data management functions required by the University. This system will be used to develop the new Employee Information System and other applications. After each system is implemented there will be post-implementation audits to determine whether MODEL 204/IFAM is performing as expected.

Other members of the staff who worked on this project include:

D. J. Farrell, A. M. Koss, E. Ekstrand, J. Noonan, G. Thurow, A. Paxson and R. Carroll.

FIGURE 1

COMPLETE DATA MANAGEMENT SYSTEM

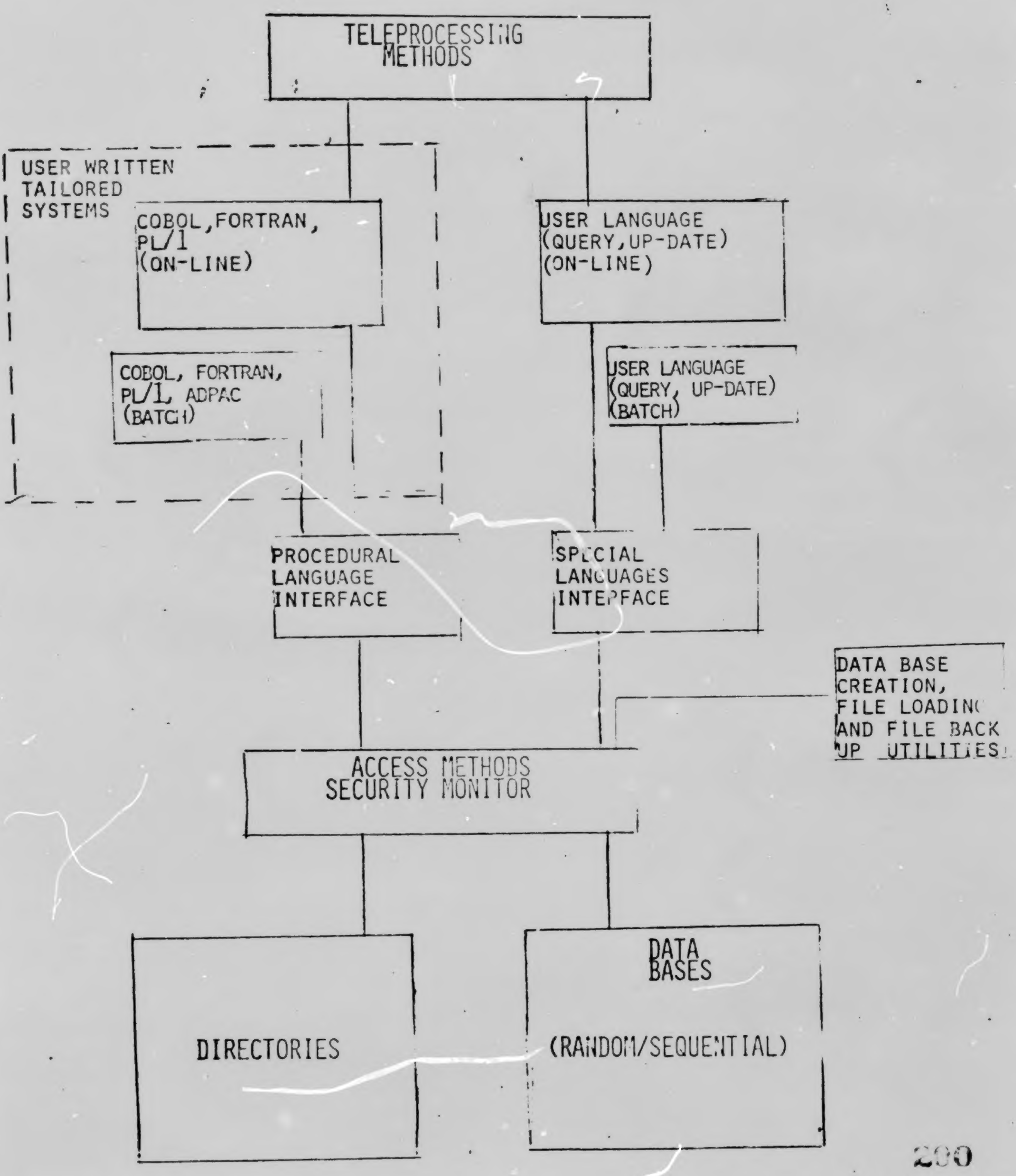

TELEPROCESSING METHODS
USER WRITTEN TAILORED SYSTEMS
COBOL, FORTRAN, PL/1 (ON-LINE)
COBOL, FORTRAN, PL/1, ADPAC (BATCH)
USER LANGUAGE (QUERY, UP-DATE) (ON-LINE)
USER LANGUAGE (QUERY, UP-DATE) (BATCH)
PROCEDURAL LANGUAGE INTERFACE
SPECIAL LANGUAGES INTERFACE
DATA BASE CREATION, FILE LOADING AND FILE BACK UP UTILITIES
ACCESS METHODS SECURITY MONITOR
DIRECTORIES
DATA BASES (RANDOM/SEQUENTIAL)

FIGURE 3

500 PERSONNEL RECORDS

LOAD	CORE	CPU TIME (MIN)	DISK ACCESSESS	COSTS 370/165 OS/ASP	DISK STORAGE (BYTES)
IMS	162K	.087	283	$2.27	117K (1 KEY)
MODEL 204/IFAM	200K	.216	2791	$5.26	173K (5 KEYS)
SYSTEM 2000	146K	.198	721	$4.19	139K (5 KEYS)
	314K	.215	607	$5.23	
TOTAL	70K	.118	2877	$6.87	179K (5 LINKAGE PATHS)

FIGURE 4

PERFORMANCE FOR 53 TRANSACTIONS

	CORE	DISK ACCESSES	CPU TIME (MIN)	COST	PROGRAMMING CC
IMS	150K	315	.015	$1.04	$600
MODEL 204/IFAM	200K	622	.118	$4.03	$240
SYSTEM 2000	138K	4551	.101	$11.18	$240
	306K	477	.035	$2.28	
TOTAL	70K	620	.079	$2.85	$600
TRANSACTIONS		INSERT DATA - 10 DELETE DATA - 7 REPLACE DATA- 7 RETRIEVE DATA-29			

FIGURE 5

SOME ADDITIONAL TESTS

1. COBOL PROGRAM (BATCH)

	CORE	CPU TIME (MIN)	DISK ACCESSESS	COST	
MODEL 204/IFAM	200K	.081	985	$3.63	
SYSTEM 2000	170K	.079	5540	$12.59	

2. USER LANGUAGE ON-LINE

	CPU COST	CORE COST	I/O COST	CONNECT COST	TOTAL
MODEL 204/IFAM	$1.30	$5.74	$2.33	$1.19	$10.57
SYSTEM 2000	$1.28	$7.16	$6.20	$1.94	$16.58

A SUGGESTED ACQUISITION PROCEDURE

John J. Sullivan
University of Rochester

YOUR CHANCES OF FINDING THE RIGHT SOFTWARE PACKAGE (DATA BASE MANAGEMENT SYSTEM OR WHATEVER) AND APPLYING IT EFFECTIVELY TO YOUR REQUIREMENTS WILL BE FAR GREATER IF YOU'RE WILLING TO TAKE THE TIME AND TROUBLE TO GO ABOUT IT IN A SYSTEMATIC, LOGICAL MANNER. THE FOLLOWING PROCEDURE, WHILE BY NO MEANS THE ONLY ONE THAT COULD RATIONALLY BE USED, IS AN EFFECTIVE, TIME-TESTED RECIPE THAT WILL VIRTUALLY GUARANTEE A SATISFACTORY SOLUTION TO YOUR SOFTWARE PROBLEMS, WHILE PROTECTING YOU FROM THE HUGE WASTES OF TIME AND MONEY THAT CAN OCCUR WHEN AN INSTITUTION PINS ITS HOPES ON AN INEFFECTUAL OR UNSUITABLE PACKAGE.

1. DETERMINE THE REQUIREMENTS.
   - A. JUST AS IN THE CASE OF IN-HOUSE SOFTWARE DEVELOPMENT, THE FIRST TASK IS TO DEFINE THE PROBLEM TO BE SOLVED AND THE ENVIRONMENT FOR ITS SOLUTION.
   - B. THE BASIC FUNCTIONS TO BE PERFORMED BY THE PACKAGE MUST BE CLEARLY DEFINED
     - 1) USUALLY IN TERMS OF THE INPUTS TO BE PROVIDED.
     - 2) THE CALCULATIONS TO BE PERFORMED
     - 3) THE OUTPUTS TO BE PRODUCED.
   - C. EQUALLY IMPORTANT, TO IDENTIFY THE MANNER IN WHICH THE PACKAGE MUST INTERFACE WITH ALL EXISTING SYSTEMS.
   - D. FINALLY, THE ENVIRONMENT IN WHICH THE PACKAGE WILL OPERATE NEEDS TO BE DEFINED
     - 1) IN TERMS OF YOUR EQUIPMENT CONFIGURATION,
     - 2) PROGRAMMING LANGUAGE, OPERATING SYSTEM, AND ANY OTHER SOFTWARE WITH WHICH THE PACKAGE MUST INTERFACE.

2. GATHER INFORMATION ABOUT THE AVAILABLE PACKAGES.

   A. THE INFORMATION REQUIRED TO SURVEY AND EVALUATE SOFTWARE PACKAGES CAN COME FROM A WIDE RANGE OF SOURCES

      1) DATAPRO
      2) AUERBACH
      3) TRADE PUBLICATIONS
      4) THE SOFTWARE SUPPLIERS THEMSELVES
      5) CONSULTANTS
      6) AND USERS OF THE PACKAGES.

   B. THE SPECIFIC TYPES OF INFORMATION YOU'LL WANT TO GATHER CAN BE CALLED "FACTORS TO EVALUATE" WHICH I WILL COVER SHORTLY.

3. NARROW THE FIELD.

   A. SOON AFTER YOU BEGIN YOUR SURVEY OF THE POTENTIALLY SUITABLE SOFTWARE PACKAGES, YOU'LL PROBABLY FIND THAT SOME OF THEM SIMPLY WILL NOT BE ABLE TO MEET YOUR REQUIREMENTS FOR ONE OR MORE CLEAR-CUT REASONS.

   B. ONE MAY REQUIRE MORE CORE STORAGE OR MORE TAPE UNITS THAN YOUR COMPUTER PROVIDES.

   C. ANOTHER MAY TURN OUT TO BE A "PAPER TIGER" THAT HAS YET TO DEMONSTRATE ITS EFFECTIVENESS IN ANY USER INSTALLATION.

   D. STILL ANOTHER MAY BE CLEARLY OVERPRICED.

   E. THESE OBVIOUSLY UNSUITABLE PACKAGES SHOULD BE QUICKLY REJECTED FROM FURTHER CONSIDERATION.

   E. FOCUS ALL OF YOUR ATTENTION ON THE REAL CONTENDERS.

4. PERFORM A DETAILED EVALUATION AND COMPARISON.
   - A. APPLY THE TESTS YOU'VE LISTED IN THE "FACTORS TO EVALUATE" TO EACH OF THE COMPETING PACKAGES.
   - B. DECIDE WHICH OF THESE FACTORS ARE MOST IMPORTANT TO YOU, IN YOUR APPLICATION, AND WEIGH THEM ACCORDINGLY.
   - C. SOME FACTORS MUST BE VIEWED AS "GO OR NO-GO" DECISION POINTS.
   - D. FOR EXAMPLE, A PACKAGE THAT WON'T BE READY FOR DELIVERY FOR SIX MONTHS AFTER YOUR DEADLINE DATE CLEARLY MUST BE DROPPED FROM FURTHER CONSIDERATION.
   - E. OTHER FACTORS ARE MUCH HARDER TO RATE OBJECTIVELY; FOR EXAMPLE, IS EASE OF INSTALLATION MORE OR LESS IMPORTANT THAN THE QUALITY OF VENDOR SUPPORT?
   - F. ONLY YOU, AS THE USER WHO WILL ULTIMATELY HAVE TO LIVE WITH THE PACKAGE OF YOUR CHOICE, CAN DECIDE.

5. TALK TO USERS.
   - A. THE PACKAGES WHICH HAVE EMERGED AS LEADING CONTENDERS FOR YOUR SOFTWARE DOLLARS SHOULD NOW BE FURTHER CHECKED OUT BY LOCATING AND CONFERRING WITH PRESENT USERS.
   - B. ASK THE SOFTWARE VENDOR FOR A LIST OF HIS CUSTOMERS. IF HE REFUSES, TAKE YOUR BUSINESS ELSEWHERE.
   - C. WHEN YOU TALK WITH A USER, REMEMBER THAT HE'S NOT REALLY LIKELY TO ADMIT HE MADE A BIG MISTAKE BY ACQUIRING A BAD PACKAGE.
   - D. ASK HIM SPECIFIC QUESTIONS ABOUT THE LIMITATIONS, DELAYS, ERRORS, AND ANY OTHER PROBLEMS HE'S ENCOUNTERED.
   - E. ASK HOW HE THINKS THE PACKAGE COULD AND SHOULD BE IMPROVED.
   - F. FINALLY, AKS FOR SPECIFIC DETAILS ABOUT THE PERFORMANCE AND THROUGHPUT HE'S EXPERIENCED.
   - G. THE FACTS AND OPINIONS GLEANED IN THIS MANNER WILL QUICKLY MAKE YOU A FAR MORE ENLIGHTENED SOFTWARE BUYER.

6. CONDUCT BENCHMARK TESTS.
   A. PREPARING AND RUNNING BENCHMARK TESTS ON YOUR OWN COMPUTER SYSTEM IS BY FAR THE MOST CONVINCING WAY TO ASSURE THE SELECTION OF A SUITABLE SOFTWARE PACKAGE. UNFORTUNATELY, IT IS NOT ALWAYS A PRACTICAL THING TO DO.
   B. DESIGNING A REALLY MEANINGFUL TEST, PREPARING THE INPUT DATA, AND EVALUATING THE RESULTS CAN BE AN EXPENSIVE, TIME-CONSUMING PROCESS.
   C. WHAT'S MORE, THE SOFTWARE VENDOR WILL OFTEN RESIST THE IDEA OF A TEST ON YOUR PREMISES BECAUSE OF BOTH THE HIGH COST TO HIM AND THE ATTENDANT SECURITY RISK.
   D. IT MAY BE REASONABLE TO WAIVE THE REQUIREMENT FOR A BENCHMARK TEST IN CASES WHERE THE PACKAGE IS WIDELY USED AND HAS RECEIVED STRONG, POSITIVE REFERENCES FROM ITS USERS.
   E. WHERE THE PACKAGE IS NEW AND UNTRIED, A CONVINCING BENCHMARK DEMONSTRATION SHOULD BE AN ABSOLUTE REQUIREMENT.

7. MAKE THE DECISION.
   A. DECIDE WHICH OF THE AVAILABLE PACKAGES WILL PERFORM THE NECESSARY FUNCTIONS AND SATISFY ALL YOUR EVALUATION CRITERIA AT THE LOWEST OVERALL COST.
   B. BE SURE TO CONSIDER NOT ONLY THE COST OF THE SOFTWARE ITSELF BUT ALL OF THE ACCOMPANYING INDIRECT COSTS:
      1) MODIFICATIONS
      2) INSTALLATION
      3) CONVERSION
      4) TRAINING
      5) MAINTENANCE
      6) DOCUMENTATION
      7) MACHINE TIME, ETC.

8. NEGOTIATE A SOUND CONTRACT.
    - A. BY ALL MEANS, RESIST THE TEMPTATION TO SIMPLY SIGN THE SOFTWARE VENDOR'S STANDARD CONTRACT.
    - B. YOU'LL NEED TO BE A FAIRLY TOUGH NEGOTIATOR TO BE SURE OF GETTING ALL THE PROTECTION AND SUPPORT YOU NEED.
    - C. THE ASSISTANCE OF YOUR COMPANY'S LAWYER MAY BE WELL WORTH HAVING.

9. INSTALL THE PACKAGE.
    - A. DEPENDING UPON THE COMPLEXITY OF THE PACKAGE AND THE ENVIRONMENT IN WHICH IT MUST OPERATE, THE INSTALLATION PHASE MAY TAKE ANYWHERE FROM ONE DAY TO MANY MONTHS.
    - B. THE VENDOR WILL USUALLY CONSIDER THE INSTALLATION TO BE COMPLETE WHEN HE
        - 1) HAS SUPPLIED THE PROMISED DOCUMENTATION
        - 2) PROVIDED THE AGREED-UPON TRAINING
        - 3) AND SUCCEEDED IN MAKING THE PACKAGE RUN ON YOUR COMPUTER.
    - C. HOWEVER, DO NOT LET THE VENDOR "OFF THE HOOK" UNTIL THE PACKAGE IS
        - 1) OPERATING AT PEAK EFFICIENCY
        - 2) HAS DELIVERED BUG-FREE RESULTS DURING AT LEAST ONE COMPLETE PROCESSING CYCLE,
        - 3) AND HAS BEEN FULLY INTEGRATED INTO THE OVERALL OPERATIONAL ENVIRONMENT.
    - D. AND REMEMBER, THERE WILL BE A CONTINUING NEED, AS WITH MOST COMPUTER APPLICATIONS, TO MODIFY AND IMPROVE THE PROGRAMS AND TO TRAIN NEW PERSONNEL IN THEIR USE.

10. CHECK THE RESULTS.

A. A SOFTWARE PACKAGE CAN REPRESENT A SIGNIFICANT CAPITAL EXPENDITURE AND A VITAL COG IN YOUR INSTITUTION'S OPERATIONS.

B. AS SUCH, ITS PERFORMANCE SHOULD BE CAREFULLY MEASURED.

C. IS IT BEING UTILIZED EFFECTIVELY?

D. IS IT DELIVERING EVERYTHING YOU BARGAINED FOR.

E. OPERATIONAL EFFICIENCIES DERIVED FROM THE USE OF THE PACKAGE SHOULD BE ANALYZED AND REPORTED IN TERMS OF MANHOURS, MACHINE-TIME REQUIREMENTS, AND OVERALL COSTS BEFORE AND AFTER INSTALLATION OF THE PACKAGE.

F. ANY SIGNIFICANT LIMITATIONS OR DEFICIENCIES IN THE PACKAGE SHOULD BE IDENTIFIED

G. AND PLANS MADE FOR OVERCOMING THEM.

H. FINALLY, A REPORT ON THE PERFORMANCE AND RELIABILITY OF THE SOFTWARE VENDOR SHOULD BE PREPARED AND FILED FOR FUTURE REFERENCE.

FACTORS TO EVALUATE

A. THERE ARE NUMEROUS CRITERIA WHICH SHOULD BE CAREFULLY CONSIDERED IN EVALUATING AND COMPARING THE MERITS OF ANY SOFTWARE PACKAGES.

B. PREPARE A LIST, ARRANGED IN THE FORM OF SPECIFIC QUESTIONS, WHICH WILL HELP YOU GUARD AGAINST THE DANGER OF OVERLOOKING ANY SIGNIFICANT FACTORS THAT MIGHT MAKE A GIVEN PACKAGE OVERLY COSTLY OR OTHERWISE UNSUITABLE FOR YOUR REQUIREMENTS.

C. YOU SHOULD DEMAND STRAIGHT-FORWARD ANSWERS FROM EACH OF THE PROSPECTIVE SOFTWARE VENDORS.

D. THEN TRY TO VERIFY AS MANY OF THE ANSWERS AS POSSIBLE.

1. DOES THE BASIC FUNCTION OF THE PACKAGE MEET YOUR NEEDS?
   A. ARE ITS OVERALL CAPABILITIES CONSISTENT WITH YOUR REQUIREMENTS AND YOUR APPLICATION?
   B. NO MATTER HOW IMPRESSIVE THE PACKAGE OR THE SALESMAN, IF IT CAN'T MEET YOUR BASIC NEEDS YOU MAY AS WELL FORGET IT.

2. WILL THE PACKAGE RUN ON YOUR COMPUTER SYSTEM?
   A. IS A VERSION AVAILABLE FOR USE ON YOUR MAKE AND MODEL OF COMPUTER?
   B. HOW MUCH CORE STORAGE IS REQUIRED?
   C. HOW MANY DISC DRIVES, TAPE UNITS, CARD READERS, PUNCHES, AND/OR PRINTERS?
   D. WHICH MODELS OF EACH?
   E. ARE ANY OPTIONAL HARDWARE FEATURES REQUIRED?
   F. HOW MANY INPUT/OUTPUT CHANNELS ARE NECESSARY?
   G. WILL ANY OFF-LINE EQUIPMENT, SUCH AS KEY-TO-TAPE DATA RECORDERS OR PUNCHED-CARD TAB MACHINES, BE NEEDED?
   H. IF ANY ADDITIONAL EQUIPMENT WOULD HAVE TO BE ADDED TO YOUR SYSTEM TO SUPPORT THE PACKAGE, ITS COST AND AVAILABILITY WILL NATURALLY NEED TO RECEIVE DUE CONSIDERATION.
   I. IF YOUR SYSTEM CONFIGURATION IS CLOSE TO THE MINIMUM FOR THE PACKAGE, MAKE SURE THAT HARDWARE LIMITATIONS WILL NOT PREVENT YOU FROM RUNNING THE PROGRAMS AT PEAK EFFICIENCY OR TAKING ADVANTAGE OF DESIRABLE OPTIONAL FACILITIES.

3. WHAT ARE THE SOFTWARE REQUIREMENTS?
   - A. CAN THE PACKAGE BE RUN UNDER THE OPERATING SYSTEM YOUR INSTALLATION NORMALLY USES?
   - B. WILL IT INTERFACE PROPERLY WITH YOUR INPUT/OUTPUT CONTROL ROUTINES AND OTHER RELATED SOFTWARE FACILITIES?
   - C. IN WHAT PROGRAMMING LANGUAGE IS THE PACKAGE WRITTEN?
   - D. ARE YOUR PROGRAMMERS FAMILIAR WITH THAT LANGUAGE?
   - E. WILL THEY FIND THE PACKAGE EASY TO UNDERSTAND, MAINTAIN AND MODIFY?
   - F. IS THE LANGUAGE SUCH THAT THE PACKAGE CAN BE MODIFIED FOR OPERATION ON YOUR NEXT COMPUTER WITHOUT UNDUE DIFFICULTY?

4. HOW WELL DO THE DETAILED CAPABILITIES OF THE PACKAGE MATCH YOUR REQUIREMENTS?
   - A. ARE ITS INPUTS, COMPUTATIONAL CAPABILITIES, AND OUTPUTS ALL CONSISTENT WITH YOUR NEEDS?
   - B. IS IT CAPABLE OF PRODUCING THE REPORTS YOU NEED? DO THE FORMATS AND CONTENTS OF ALL THE INPUT DOCUMENTS, MASTER FILES, AND REPORTS FIT YOUR NEEDS?
   - C. OR CAN THEY BE MADE TO DO SO WITHOUT UNDUE COST AND DIFFICULTY?
   - D. ARE THE FILE ORGANIZATION AND DATA MANAGEMENT TECHNIQUES SATISFACTORY?
   - E. WHAT INPUT/OUTPUT AND PROCESSING OPTIONS ARE AVAILABLE, IF ANY?
   - F. WILL SPECIAL FORMS BE REQUIRED?
   - G. WHAT CONTROL PROCEDURES AND AUDIT TRAILS ARE INCORPORATED?
   - H. WHAT TYPES OF ERROR CHECKS ARE MADE, AND WHAT HAPPENS WHEN ERRORS ARE DETECTED?
   - I. WHAT PROVISIONS ARE MADE FOR RERUNS?
   - J. WHAT ABOUT FILE PROTECTION AND DATA SECURITY?
   - K. ONLY YOU CAN JUDGE WHETHER A GIVEN PACKAGE MEETS--OR CAN BE ECONOMICALLY ADAPTED TO MEET--YOUR REQUIREMENTS IN EACH OF THESE IMPORTANT AREAS.

5. IS THE PERFORMANCE ADEQUATE FOR YOUR NEEDS?
   - A. SOLID, BELIEVABLE PERFORMANCE FIGURES FOR SOFTWARE PACKAGES ARE STILL HARD TO COME BY--BUT ESSENTIAL TO HAVE BEFORE PLACING YOUR ORDER.
   - B. PACKAGES WITH SIMILAR FUNCTIONS AND EQUIPMENT REQUIREMENTS CAN VARY GREATLY IN THROUGHPUT.
   - C. HARDWARE FACTORS SUCH AS MEMORY SIZE AND TAPE SPEED CAN HAVE MAJOR EFFECTS ON PERFORMANCE.
   - D. A GOOD INFORMATION SOURCE IS CURRENT USERS--BUT MAKE SURE THEIR EQUIPMENT CONFIGURATIONS AND OPERATIONAL ENVIRONMENTS ARE COMPARABLE WITH YOUR OWN.
   - E. THE VERY BEST SOURCE OF PERFORMANCE DATA IS BENCHMARK TESTS ON YOUR OWN SYSTEM--BUT BENCHMARK TESTING IS AN EXPENSIVE, TIME-CONSUMING PROPOSITION.
   - F. IT SHOULD BE DELAYED (AVOIDED IF POSSIBLE) UNTIL THE FIELD HAS BEEN NARROWED TO THE TWO OR THREE STRONGEST CONTENDERS.

6. HOW FLEXIBLE IS THE PACKAGE?
   - A. ARE ITS INPUT, OUTPUT AND PROCESSING CAPABILITIES FLEXIBLE ENOUGH TO ACCOMMODATE CHANGING REQUIREMENTS.
   - B. DOES IT INCLUDE I/O AND COMPUTATIONAL CAPABILITIES BEYOND YOUR PRESENT NEEDS, WHICH MAY BE DESIRABLE IN THE FUTURE?
   - C. ARE THE RECORD LENGTHS AND FILE CAPACITIES LARGE ENOUGH TO MEET YOUR REQUIREMENTS FOR THE FORESEEABLE FUTURE?
   - D. CAN THE SYSTEM BE EXPANDED AND/OR MODIFIED EASILY TO SATISFY THE NEEDS OF THE FUTURE AS WELL AS THE PRESENT?
   - E. (IT SHOULD BE NOTED THAT THIS TYPE OF FLEXIBILITY, WHILE HIGHLY DESIRABLE, USUALLY RESULTS IN A HIGHER PRICE TAG AND LOWER OPERATIONAL EFFICIENCY FOR THE PACKAGE.)

7. IN WHAT FORM IS THE PACKAGE DELIVERED?
   A. FOR SECURITY REASONS, SOME PACKAGES ARE MARKETED ONLY IN OBJECT-LANGUAGE FORM.
   B. WHENEVER POSSIBLE, HOWEVER, YOU SHOULD DEMAND A SOURCE DECK AND SOURCE-LANGUAGE LISTING.
   E. THESE MAKE IT FAR EASIER TO UNDERSTAND THE INNER WORKINGS OF THE PACKAGE, UTILIZE IT EFFECTIVELY, AND MODIFY IT WHEN NECESSARY.

8. HOW DIFFICULT WILL IT BE TO INSTALL THE PACKAGE?
   A. WHAT CHANGES WILL NEED TO BE MADE IN YOUR EXISTING SYSTEMS, PROCEDURES, AND FORMS?
   B. HOW MANY PEOPLE WILL BE AFFECTED, AND WHAT TYPE OF TRAINING OR ORIENTATION WILL THEY NEED?
   C. WILL DATA FILES NEED TO BE CONVERTED TO NEW FORMATS AND/OR MEDIA?
   D. IF SO, WHEN AND HOW?
   E. HOW MUCH SPECIALIZED DOCUMENTATION, IN ADDITION TO THAT SUPPLIED BY THE PACKAGE VENDOR, WILL NEED TO BE PRODUCED?
   F. IS IT PRACTICAL TO SCHEDULE A PERIOD OF PARALLEL OPERATION, WITH A GRADUAL CUT-OVER FROM THE OLD SYSTEM TO THE NEW ONE?
   G. CAN THE VENDOR FURNISH A RECOMMENDED PROCEDURE FOR THE INSTALLATION AND CONVERSION PROCESS?
   H. HOW MUCH ASSISTANCE WILL THE VENDOR FURNISH DURING THE PROCESS?
   I. AT WHAT PRICE?
   J. WHILE ASKING THESE QUESTIONS, KEEP IN MIND THAT THE COSTS ENCOUNTERED IN INSTALLING A PACKAGE OFTEN FAR EXCEED THE DIRECT COST OF THE PACKAGE ITSELF.

9. WILL THE PACKAGE BE EASY TO USE?

   A. IS IT DESIGNED FOR STRAIGHT-FORWARD OPERATION ON YOUR COMPUTER SYSTEM, WITH WELL-DOCUMENTED OPERATING PROCEDURES?

   B. ARE THE INPUT FORMS AND THE INSTRUCTIONS FOR PREPARING THEM CLEAR AND EFFICIENT?

   C. ARE THE REPORTS IT PRODUCES EQUALLY CLEAR, COMPREHENSIVE, AND SELF-EVIDENT?

   D. WILL EVERYONE WHO IS AFFECTED BY THE PACKAGE FIND THAT IT SATISFIES HIS NEEDS WITH A MINIMUM OF EFFORT ON HIS PART?

10. IS ADEQUATE DOCUMENTATION PROVIDED?

    A. GOOD, COMPREHENSIVE DOCUMENTATION, IMPORTANT IN EVERY COMPUTER INSTALLATION, IS ABSOLUTELY VITAL IN THE CASE OF PROPRIETARY PACKAGES, SINCE THE WRITERS OF THE PROGRAMS WILL SELDOM BE AVAILABLE WHEN QUESTIONS AND PROBLEMS ARISE.

    B. A CAREFUL CHECK ON THE QUANTITY AND QUALITY OF DOCUMENTATION IS ONE OF THE QUICKEST AND MOST RELIABLE WAYS TO JUDGE THE OVERALL QUALITY OF A PACKAGE.

    C. IN GENERAL, FOUR DIFFERENT LEVELS OF DOCUMENTATION SHOULD BE PROVIDED, ORIENTED TOWARD THE RESPECTIVE NEEDS OF THE SYSTEMS ANALYST, THE MAINTENANCE PROGRAMMER, THE COMPUTER OPERATOR, AND THE PACKAGE USER.

    D. SPECIFIC ELEMENTS OF DOCUMENTATION THAT SHOULD BE AVAILABLE INCLUDE A COMPREHENSIVE USER'S MANUAL, NARRATIVE DESCRIPTIONS, SYSTEM AND PROGRAM FLOWCHARTS, SOURCE PROGRAM LISTINGS, INPUT DOCUMENT LAYOUTS, INTERNAL RECORD LAYOUTS, REPORT LAYOUTS, OPERATING INSTRUCTIONS, AND INSTRUCTIONS FOR PREPARING THE INPUT AND UTILIZING THE OUTPUT.

11. WHAT SUPPORT WILL THE VENDOR PROVIDE?
    - A. HOW MUCH ON-SITE TECHNICAL ASSISTANCE IS PROVIDED UNDER THE VENDOR'S STANDARD CONTRACT?
    - B. WHAT SPECIFIC QUALIFICATIONS WILL THESE PEOPLE HAVE?
    - C. WHAT IS THE COST AND AVAILABILITY OF ADDITIONAL ASSISTANCE IF REQUIRED?
    - D. HOW MUCH TRAINING WILL BE SUPPLIED FOR YOUR PERSONNEL?
    - E. WHAT IS THE NATURE AND LEVEL OF THE TRAINING COURSES, AND WHERE AND WHEN ARE THEY CONDUCTED?
    - F. WILL THE VENDOR MAKE ANY MODIFICATIONS THAT MAY BE NECESSARY TO ADAPT THE PACKAGE TO YOUR SPECIFIC NEEDS?
    - G. AT WHAT ADDITIONAL COST, IF ANY?
    - H. WILL THE VENDOR CORRECT, WITHOUT CHARGE, ANY BUGS THAT ARE DISCOVERED AFTER INSTALLATION OF THE PACKAGE HAS BEEN COMPLETED?
    - I. FOR HOW LONG?
    - J. WILL THE VENDOR SUPPLY IMPROVED VERSIONS OF THE PACKAGE FROM TIME TO TIME?
    - K. WHAT CHARGE, IF ANY, IS IMPOSED FOR CONTINUING MAINTENANCE?

12. WHAT IS THE OPERATIONAL STATUS OF THE PACKAGE?

    A. WHEN, WHERE, BY WHOM, AND FOR WHAT PURPOSE WAS IT ORIGINALLY DEVELOPED?

    B. WHEN WAS THE FIRST INSTALLATION COMPLETED FOR A USER OTHER THAN THE DEVELOPER?

    C. HOW MANY COMPANIES ARE CURRENTLY USING THE PACKAGE?

    D. ONE OF THE BIGGEST POTENTIAL BENEFITS OF A SOFTWARE PACKAGE IS THE HOPE THAT IT HAS ALREADY BEEN TESTED, DEBUGGED, AND PROVEN EFFECTIVE IN NUMEROUS OTHER INSTALLATIONS.

    E. IN MANY CASES THIS IS TRUE. BUT IN MANY OTHER CASES THE UNWARY BUYER WILL COMMIT HIMSELF TO A PACKAGE THAT IS STILL IN THE EARLY STAGES OF DEVELOPMENT, WITH THE ACCOMPANYING LIKELIHOOD OF COSTLY DELAYS AND DISILLUSIONMENTS.

    F. AT PRESENT, ONE OF THE BEST TESTS OF THE QUALITY AND EFFECTIVENESS OF A SOFTWARE PACKAGE IS THE NUMBER OF SATISFIED USERS--AND YOU CAN EASILY FIND OUT HOW SATISFIED THEY ARE BY DEMANDING THEIR NAMES AND CONTACTING THEM YOURSELF.

13. WHAT IS THE TOTAL COST OF ACQUIRING AND USING THE PACKAGE?

    A. THIS INVOLVES CAREFUL CONSIDERATION OF BOTH THE DIRECT COST (I.E., THE PRICE OF THE PACKAGE ITSELF) AND THE INDIRECT COSTS THAT MAY BE INCURRED IN MODIFYING THE PACKAGE TO FIT YOUR REQUIREMENTS

        1) CHANGING EXISTING SYSTEMS AND PROCEDURES
        2) TRAINING YOUR PERSONNEL
        3) CONVERTING YOUR FILES
        4) INSTALLING AND CHECKING OUT THE PACKAGE
        5) OPERATING THE PACKAGE ON A PRODUCTION BASIS
        6) AND MAINTAINING IT AFTER INSTALLATION.

    B. NOTE THAT THESE INDIRECT COSTS MAY VARY CONSIDERABLY AMONG COMPETITIVE PACKAGES BECAUSE OF DIFFERENCES BOTH IN THE PACKAGES THEMSELVES AND IN THEIR VENDORS' SUPPORT POLICIES.

14. WHAT FINANCIAL ARRANGEMENTS ARE OFFERED?
    A. IS THE PACKAGE AVAILABLE FOR OUTRIGHT SALE, FOR LEASE, OR BOTH?
    B. WHAT ARE THE SPECIFIC TERMS AND CONDITIONS FOR EACH PLAN?
    C. ARE THERE ANY OBJECTIONABLE CONSTRAINTS UPON THE USE OR MODIFICATION OF THE PACKAGE, OR UPON THE SALE OF SERVICES BASED ON ITS USE?
    D. IS A DISCOUNT OFFERED FOR MULTIPLE INSTALLATIONS WITHIN THE SAME INSTITUTION (OR EDUCATION)?
    E. ARE THERE ANY EXTRA-COST OPTIONS?
    F. IF SO, ARE THEY WORTH THE ASKING PRICE, AND CAN THEY BE ADDED AT A LATER DATE?
    G. WHAT ADDITIONAL CHARGES ARE IMPOSED FOR SUPPORT, TRAINING, OR CONTINUED MAINTENANCE?

WHEN YOU'VE GATHERED, ANALYZED, AND COMPARED THE ANSWERS TO THESE QUESTIONS FOR EACH OF THE PACKAGES UNDER CONSIDERATION, YOU'LL BE IN A POSITION TO MAKE A TRULY INFORMED, CONFIDENT DECISION. IT'S A TIME-CONSUMING PROCESS, OF COURSE, BUT IF YOU'RE LOOKING FOR LONG-TERM ECONOMY AND SATISFACTION, YOU REALLY CAN'T AFFORD TO NEGLECT THIS TYPE OF CAREFUL, COMPREHENSIVE EVALUATION.

PACKAGE NAME	1	2	3	4	5
Average memory used					
Overall satisfaction					
Excellent					
Good					
Fair					
Poor					
Grade point average					
Throughput/efficiency					
Excellent					
Good					
Fair					
Poor					
Grade point average					
Ease of installation					
Excellent					
Good					
Fair					
Poor					
Grade point average					
Documentation					
Excellent					
Good					
Fair					
Poor					
Grade point average					
Vendor technical support					
Excellent					
Good					
Fair					
Poor					
Grade point average					
Perform as advertised?					
Immediately					
Eventually					
Never					
Require modification?					
No					
Yes, by vendor					
Yes, by user					

Five subjective rating questions could be asked, with the answers "excellent," "good," "fair," "poor" available to be checked. The questions: "How would you rate the package with respect to (1) your overall satisfaction, (b) job throughput/program efficiency, (c) ease of installation, (d) documentation, and (e) vendor's technical support?" Following the subjective questions, the questionnaire could contain two objective checkoff questions: "Did the package perform as advertised -- immediately, eventually, or never?" and "Did the package require modification -- not at all, by the vendor, or by the user?" You must make up comparison charts on the packages reported on. These charts are instructive in that they show the raw rating tallies and a computed "grade point average" for each subjective rating category. The grade point averages could be computed in a manner familiar to most. Excellent is weighted as 4, good as 3, fair as 2, and poor as 1. Then, the tallied number of each response, multiplied by the corresponding weight, and an average taken by dividing the sum of the products by the number of responses totalled.

DATA BASE

THE CONCEPT

THE COMMITMENT

"It is a capital mistake to theorize before one has data."
Sir Arthur Conan Doyle

"Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it."
Samuel Johnson

Prepared for Presentation
By R. Brian Walsh
CAUSE -- December 1973
Data Base Management Evaluation Panel

FORWARD

I am deeply indebted to Miss Valissa Hilligoss, Manager of Data Base Systems at the University of Notre Dame du Lac. Her unpublished reports and patient assistance gave me insight into the concept of data base and enabled me to prepare this paper for presentation.

DATA BASE: THE CONCEPT AND THE COMMITMENT

I. FORWARD Page i

II. INTRODUCTION Page 1

III. MANAGEMENT INFORMATION SYSTEM Page 4

IV. DATA BASE CONCEPT Page 10

V. DATA BASE MANAGEMENT SYSTEMS Page 16

VI. THE DECISION Page 18

VII. THE COMMITMENT Page 21

EXHIBITS

Levels of Information (Figure 1) Page 7

Administrative Information Flow and Data Bases (Figure 2) Page 8

Application Dependence (Figure 3) Page 13

INTRODUCTION

Universities and colleges are complex business organizations composed of multiple interrelated operational units. Communication lines between these units are often poorly defined and difficult. Increasing demands for information concerning university operations and goals and a demand for tighter internal controls are being made by the institution's constituents. As a result of these pressures, educational managers are finding that they are being held accountable for their actions and/or lack of them and are being forced to plan and to make those plans available to their constituents.

It is not surprising then that university and college administrators have turned to various management groups, computer personnel, consultants and others for guidance. Their advice is to "manage" the institution by planning programs and growth, setting priorities and controlling the institution's finances. In order to accomplish this, they have been told to develop "Management Information Systems" as quickly as possible.

The emphasis to develop "Management Information Systems" has led to wide use of this term, yet it is one of the most misunderstood terms in university and college management circles; this confusion is not limited to management but also computer personnel. To some, the term means the answer to all management problems, to others, it is the ultimate in data processing support, and to most it means a highly automated, computerized system. In my opinion, a Management Information System is one that supports managerial decision-making by supplying relevant information when required. Whenever discussions regarding "Management Information Systems" arise they inevitably lead to the topic of "data base system" support for them. Many books, articles and special papers have been written on these subjects with almost as many different definitions of terms and approaches to implementation.

Among computer personnel frequently the terms "data base systems" and "data base management systems" are considered to be interchangeable terms. They are not. The approach to data base and choice of a data base management system must fit the overall concept and commitment of an institution to its own particular "Management Information System." The development of such a system and the data base necessary to support it requires a major commitment on the part of college or university administrators. Institutional management must be directly involved in defining requirements for information, information flows, operational departments, controls, planning and decision-making models, etc., and the set of common data elements needed to satisfy those requirements.

The purpose of this presentation is to define the previously discussed concepts with respect to their practical application. Further, management information systems, information flows, and data bases will be considered in light of their current status at the University of Notre Dame du Lac and their potential role in and impact on the University.

MANAGEMENT INFORMATION SYSTEM

Before discussing data base or data base management systems it is essential that at least a brief overview of M. I. S. be presented. The purpose of a Management Information System is to support managerial decision making by supplying relevant information when required. James McKeever, in his book Management Reporting Systems, defines M. I. S. as "a collection of facts, procedures, people and machines which develop information designed to serve as a basis for decision making and policy formation." A M. I. S. does not make decisions, it provides management with the necessary data to aid it in its process of decision making. A M. I. S. need not be automated! Only where it is justified should any part of or the entire system be processed on a computer.

To be of any real service to management, the management information system must be as simple as possible; it must be reliable; it must provide the data necessary to make a decision whenever it can be reasonably obtained, and it must provide that information on a timely basis. A M. I. S. must of necessity be personalized to the institution and to the managers it serves.

In serving top management needs, a management information system will cross organizational lines and organized information flows. Flexibility and growth potential are the keys to a usable M. I. S. By providing consistent data and valid reports, a management information system allows for evaluating alternative paths of action, for better documentation of resource utilization, and for coordination of all functional units of the institution. The result: sounder management! Because a M.I.S. is designed for management, an understanding of the following points must exist·

- A M.I.S. is a specialized tool for specific managers. It must be developed on the basis of executive requirements and involvement.
- The criteria by which a M.I.S. should be measured are the results and not just the costs.
- Top management must be committed to its M.I.S. They must "believe" in it!

Before considering implementation of a management information system, management must:

- Realize it has an information problem
- Want to solve its information problem
- Be willing to use the information in its decision making process
- Be willing to support the M.I.S. effort with:
    - a) Time and personal involvement
    - b) Manpower and facilities
    - c) Adequate funding

Implementation of any system does not just happen! First, decide what you want the system to accomplish. Those decisions must be documented so they are clear to those who are to implement them. In other words a plan must be developed and then executed. Such a procedure demands the following steps:

- o Formation of system design objectives and approval of those objectives stated clearly and positively by all affected parties -- particularly top management.
- o Determination of basic information requirements necessary to achieve the design objectives.
- o Design of the M.I.S. Data Base
- o Software design and/or acquisition
- o Hardware modification and/or selection
- o Program development and testing
- o A review must be made of the completed project to be sure it satisfies its design objectives.

There are three basic levels of information (see figure 1) needed for a management information system to function effectively:

- o Strategic information for planning and decision making
- o Tactical information for implementing decisions and policies and for measuring and controlling the results
- o Operational information needed to carry out the day to day procedures

These levels of information correspond to the three levels and functional flows presently defined for the University of Notre Dame du Lac (see figure 2). Information within these levels consist of actual data or results derived by manipulating or calculating the actual data. Serious

LEVELS OF INFORMATION

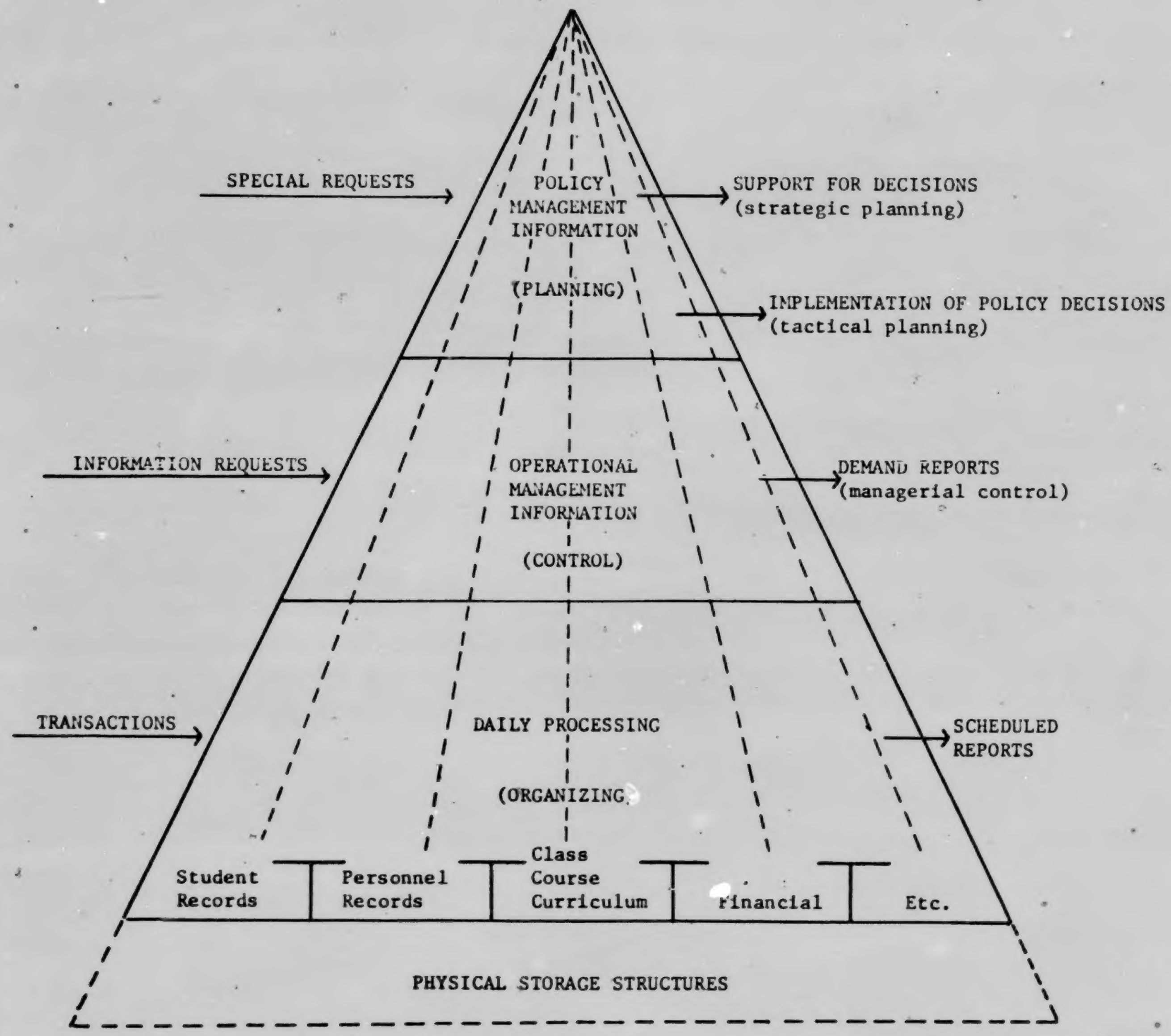

SPECIAL REQUESTS
POLICY MANAGEMENT INFORMATION
(PLANNING)
SUPPORT FOR DECISIONS (strategic planning)
IMPLEMENTATION OF POLICY DECISIONS (tactical planning)
INFORMATION REQUESTS
OPERATIONAL MANAGEMENT INFORMATION
(CONTROL)
DEMAND REPORTS (managerial control)
TRANSACTIONS
DAILY PROCESSING
(ORGANIZING
SCHEDULED REPORTS
Student Records
Personnel Records
Class Course Curriculum
Financial
Etc.
PHYSICAL STORAGE STRUCTURES

FIGURE 1

ADMINISTRATIVE INFORMATION FLOW AND DATA BASES

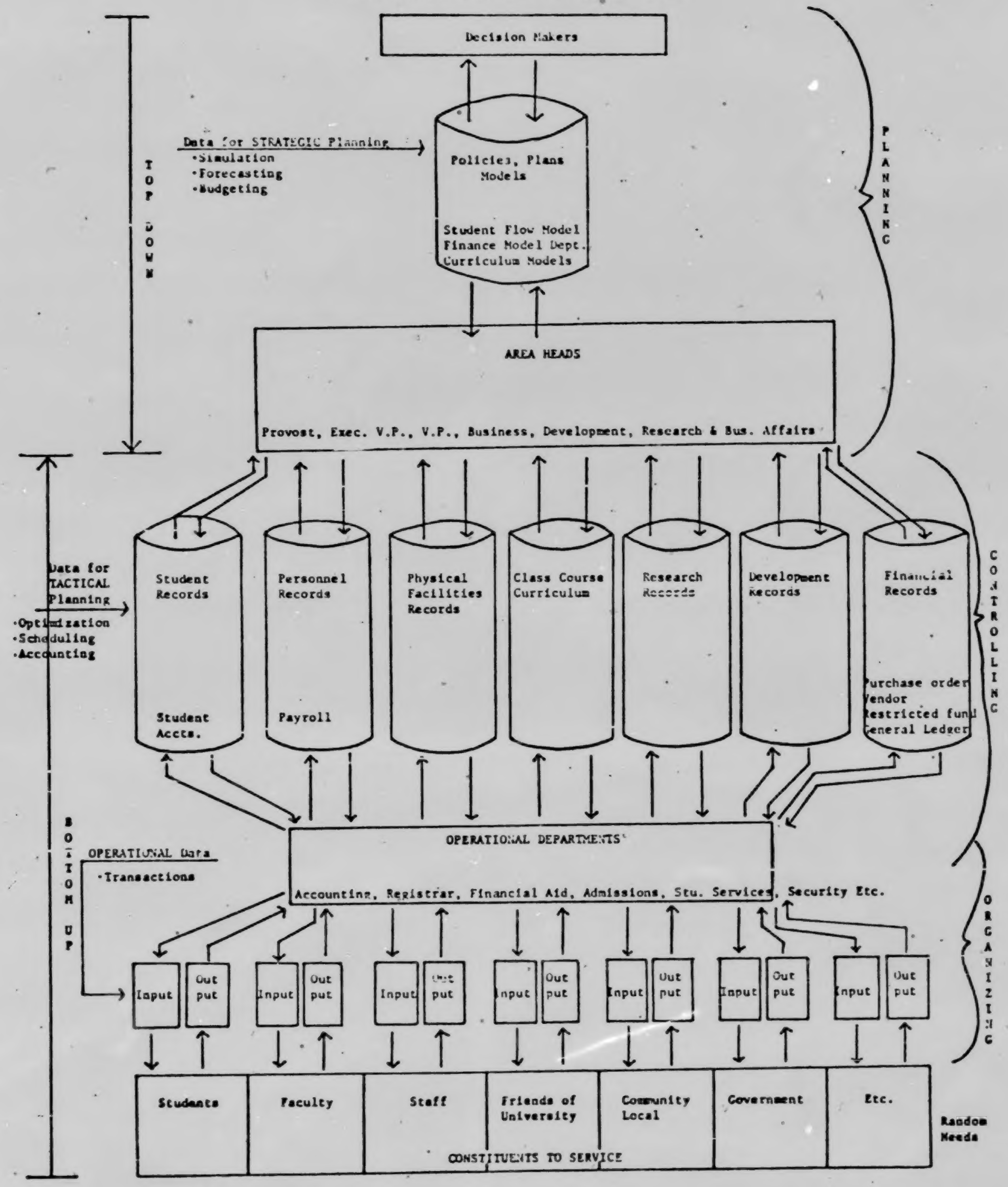

Decision Makers
Data for STRATEGIC Planning
·Simulation
·Forecasting
·Budgeting
Policies, Plans Models
Student Flow Model
Finance Model Dept.
Curriculum Models
TOP DOWN
PLANNING
AREA HEADS
Provost, Exec. V.P., V.P., Business, Development, Research & Bus. Affairs
Data for TACTICAL Planning
·Optimization
·Scheduling
·Accounting
Student Records
Student Accts.
Personnel Records
Payroll
Physical Facilities Records
Class Course Curriculum
Research Records
Development Records
Financial Records
Purchase order
Vendor
Restricted fund
General Ledger
CONTROLLING
BOTTOM UP
OPERATIONAL Data
·Transactions
OPERATIONAL DEPARTMENTS
Accounting, Registrar, Financial Aid, Admissions, Stu. Services, Security Etc.
ORGANIZING
Input
Out put
Students
Faculty
Staff
Friends of University
Community Local
Government
Etc.
Random Needs
CONSTITUENTS TO SERVICE

Figure 2

consideration must be given to the operating departmental structure so that it might efficiently gather, collate, classify and store data for use. No one organizational structure is correct for every institution. Because of the changing management techniques in universities and colleges, traditional structures may no longer be valid! A great deal of thought must be given to this to determine whether existing organizational structures are effective.

The choice of what data to collect and/or keep and the repository of actual data is its data base. This serves as the foundation for a management information system. To effectively and efficiently serve information needs, it is essential that a common and controlled approach be used in creating and manipulating physical data (i. e. a data base management system). Management information systems depend on a data base management system for building, restructuring and accessing information from the data base. It is worth noting at this point that this data base management system can be a manual system but with large volumes of data it is usually an automated system. It should be kept in mind that any serious discussion of data base or data base management systems must take into account that it is an essential part, but just one part, of a management information system.

DATA BASE CONCEPT

In the overview of a management information system, a framework was provided that will now facilitate a detailed discussion of the data base concept. That framework has shown that the data base concept is the foundation of a management information system. But what exactly is a data base? The dictionary defines data and base in the following manner:

- Data -- is something that is given either from being experientially encountered or from being admitted or assumed for a specific purpose: a fact or principle granted or presented: something upon which an inference or an argument is based or from which an intellectual system of any sort is constructed.

- Base -- is the bottom of something considered as its support: that on which something rests or stands: the fundamental part of something: basic principle.

These definitions provide us with a clue to what a data base is. Authors have variously defined data base as:

- "Data base is a collection of related information supporting business needs." by Cuozzo and Kurtz in their article, "Building a Data Base: A Management Perspective." Datamation, October 1973.

- "Data base refers to the sum total of information collected for any particular purpose to be used for implication analysis and for archival purposes" by Ralph Bisco, Data Bases, Computers and the Social Sciences. 1970.

- "Data base consists of all the record occurrences, set occurrences and areas which exist in secondary storage and are controlled by a specific scheme" by the Data Base Task Group Report to Codasyl, October 1969.

- "Data base is a set of files maintained by the system for use in user-specified creation, updating and interrogation process. The files of a data base are accessed through names or other identifiers supplied by the user in a prior definition process," in Codasyl -- Feature Analysis of Generalized Data Base Management Systems, May 1971.

I will not add a further definition to the many that already exist but in general the definition given by Codasyl in 1971 will be used.

Administrators have only recently become aware of the fact that data is a major resource. Every area of our institutions, administrative, instructional or research, depend on data. Once lost, items of data are essentially not replaceable. Richard Nolan in his article "Computer Data Bases: The Future is Now," Harvard Business Review, in the September/October 1973 issue pointed out that out of this recent awareness two key concepts of an institutional wide data base have emerged:

- "The data that computer programs use are considered independent resources in themselves, separate from the computer programs."

- "There is an art and approach to managing and structuring a company's computer - readable data as a whole, so that they constitute a resource available to the organization for broadrange applications - especially on an ad hoc basis."

The distinction between a traditional file system and a generalized data base system is subtle but of great importance. It is important because data base implementation represents an extensive institutional commitment in terms of top management participation in documenting information requirements, manpower, money, computer time and resources. It is this commitment that will eventually allow management to get information on an ad hoc basis!

A data base system is, by definition, data oriented while a file system has traditionally been function oriented (i.e. specific to programs and applications). (See figure 3). File systems represent a dead end in terms of the data whereas a data base system is capable of growth and can be used in new problem solutions, new applications and in creation of or comparison with other data.

Historically, programs have been written to fulfill specific requests. Frequently these programs have created their own special files used and arranged in a way that was unique to solving a specific problem. Consequently, after many such special applications have been developed, common data spreads throughout various files leading to redundancy and multiple definitions of data elements. New programs usually mean creating new files or preprocessing existing files to achieve the required data format. This traditional approach has undercut and in some cases aborted the advances in computer technology. It has also obstructed top management's demand for applications that require information on an ad hoc basis.

The data base system concept, on the other hand, is data oriented instead of function oriented. A data base consists of common data elements

"APPLICATION DEPENDENCE"

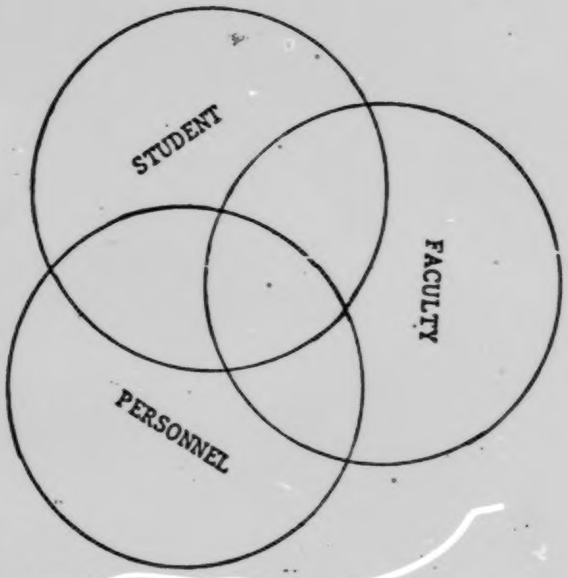
Student File

Faculty File

Personnel File

Student Directory File

Payroll File



Many interrelationships exist:

1. Payroll records for personnel include all members of the faculty, some students and staff personnel
2. Some faculty members are also students, and vise-vers
3. Student directory "file" information should be a subset of the student data; therefore, considerable redundant data is present
4. Similar data in the different files is not compatible
5. There is an overlap in processing requirements
6. The same data element may appear in multiple files and the information be different

Figure 3

for the solution of multiple problems. A data base is designed for flexibility, generality and growth. A data base becomes a valuable information source for use throughout the institution. It is the basic building block for the development of a management information system! Flexibility and the ability to grow lead to the development of those new applications which will eventually provide tremendous benefits and some savings to the institution many times beyond the original cost of implementation.

Implementation of the data base concept will allow an institution to exercise central control over one of its most valuable resources -- data. This means that the college or university must look at its total information requirements rather than at its needs for individual applications. A reasonable approach to the implementation of the data base concept will result in the construction of an institutional data bank containing nonredundant, up-to-date data elements which are accepted by the college and university community. Increased productivity should be possible because the data base lends itself to growth and makes it easier to add new reports when the need arises. Central control of a data base encourages common definitions of data which will save programming time and will allow for tighter security procedures. It provides a central point for revising and cleaning up data definitions and establishes data structures according to the natural relationships defined by the institution's operating environment.

The data base concept can be implemented in many different ways. Most manual systems are not responsive enough. Many medium to large-scale computer installations are currently evaluating or implementing a Data Base Management System (D. B. M.S.). The selection of the proper such system depends to a great extent on the type of computer resources an institution has and upon its philosophy regarding its management information system.

DATA BASE MANAGEMENT SYSTEMS

A data base management system (DBMS) is a structured series of generalized system programs that acts as a common interface point to intercept and satisfy data base information requests. It is a tool to aid in the definition, creation, maintenance and retrieval of information in the university data base. This system performs two primary functions:

- It stores and maintains the representations of structured information.
- It presents structured output information from the stored representations on request.

The data base management system provides a set of complex programs and utilities, usually supplied by an outside vendor, which support the operational development and implementation of the data base concept. These programs are really an extension of the computer's internal control system with respect to data management capability; they offer a generalized and improved approach to data manipulation.

Miss Valissa Hilligoss, Manager of Data Base Systems at the University of Notre Dame du Lac, established the following list of facilities that a data base management system should provide:

- A format for structuring data.
- A means to define this structure in terms of physical and logical attributes.
- A method to create the data base and, once created, update and delete the stored information.

- An easily implemented capability for expanding the data base in the total number or volume of records, as well as the addition of new types of information.
- A mechanism for retrieving information for reports.
- A controlled and secure environment with data protection and integrity.
- An automated system for recovery.
- Compatibility with technological advances in software and hardware.
- Data independence (logical versus physical).
- Data relatability.
- Nonredundancy.
- Data definition and access external to the program and programmer control.
- A system compatible with existing languages.
- Ease of use.
- Processing efficiency.
- Capability for dynamic restructuring to meet new user information requirements with little or no impact on existing data base applications.
- A dictionary as a data inventory system.
- Vendor support, education and documentation.

THE DECISION

A basic management information system philosophy was being formalized at the University of Notre Dame du Lac by early 1971 and as a result goals were established to develop a series of appropriate (see figure 2) data bases. Having participated in the development of this philosophy, I appointed a committee composed entirely of Computing Center staff members to investigate and select a data base management system that met as many of the criteria established for an ideal DBMS as possible. It was my hope that a reasonable selection and implementation of a DBMS would result in at least the following benefits:

- A reduction of data storage costs.
- Provide data on a timely basis.
- A reduction of data entry costs.
- Provide "secure" access to privileged data.
- Elimination of concurrent data update conflicts.
- Provide optimal performance and offer greater access flexibility.
- Unification of data recovery and back-up functions.
- Increased productivity of the programming staff through automated aids.
- The establishment of a discipline for current and future application designs.
- The establishment of a centralized control function --the data base manager.

- A reduction in cost of program maintenance.
- It would enable the Center to react to a dynamically changing environment.

With these criteria in mind, the committee reviewed over forty-eight alleged data base management systems. The results of this initial work indicated that only two true data base management systems were available at that time: IBM Corporation's "Information Management System" (IMS) and Cincom's "TOTAL" package.

After establishing criteria for evaluation, a six month investigation of the systems took place. This included several vendor presentations as well as personal contacts with users. A report of the findings set forth strong differences of opinion regarding the technical capabilities of each system. It was determined that either system could be made to work! As a result of this study, the IMS data base management system was selected in late 1971 as being the most suitable for Notre Dame's environment and M.I.S. philosophy. Reasons for this decision include but were not limited to the following:

- Long-range enhancement capability (due to the size and financial commitment of the vendor).
- A wider range of supportive utilities.
- Lower initial cost.
- Good back-up and recovery procedures.
- Sequential processing capabilities.

- Local vendor support.
- Less difficulty in determining vendor responsibility for system bugs.

IMS is strong in all of the desired DBMS facilities mentioned earlier except in the area of security (not really implemented in batch), true data independence, dynamic restructuring capability without impact and a dictionary/directory. However, no existing data base package possesses all of the facilities which were defined as being required in the ultimate system. A decision was also made to implement only the batch version of IMS and that all on-line (not to be equated with real-time in all cases) applications would be implemented via TSO. This was and is Notre Dame's time-sharing system. IMS is an extremely powerful tool for accomplishing the University of Notre Dame du Lac's goal: the creation of a reasonable data base for the formation of a sound Management Information System.

THE COMMITMENT

A brief statement regarding the word commitment as used in this presentation is required. What is really meant by commitment?

Without question, top administrators in colleges and universities are aware of the need to develop a coordinated and controlled environment. They realize that information must be available to assist in their decision-making process. They are aware that this information comes from multiple operating units and crosses organizational lines of authority. The need for a common data base and the need to define the elements of information stored in that base is clear.

There is also little doubt that administrators will provide financial support (if it can be obtained), memos, directives and at least a nodding approval. This is not enough!

If the goal of a M. I. S. is to be achieved, top administrators must involve themselves personally and they must put forth their own individual efforts to make it happen.

Personal involvement means doing everything one can to understand the concept and the efforts necessary to implement a management information system and, in particular, its data base. It means doing everything one can to develop coordination and cooperation between and/or among related operating units. It means defining in one's own mind exactly what the information requirements of an area are, not only today, but also in the future.

It means evaluating how data from one area may impact and/or interface with another area of the institution. It means documenting these answers!

Personal involvement means active participation. Each top administrator is a key to the success of the management information system and its supporting data base.

SOURCE MATERIAL AND SUPPLEMENTARY READING

Bisco, Ralph L., Data Bases, Computers and the Social Sciences. Wiley-Interscience, 1970.

CODASYL, Data Base Task Group Report to the Codasyl Programming Language Committee. October, 1969.

CODASYL, Feature Analysis of Generalized Data Base Management System, May, 1971.

Cuozzo, D. E. and Kurtz, J. F., "Building a Base for Data Base: A Management Perspective," Datamation (October, 1973)

Guthrie, David R., "A Data Dictionary Approach to MIS," Datamation (April, 1973).

Lyon, John K., An Introduction to Data Base Design. Wiley-Interscience, 1971.

Mays, Donald L., "Sound Systems Renew Hope for Education," Journal of Systems Management (November, 1973).

McKeever, James M., Management Reporting Systems. Wiley-Interscience, 1971.

Meiners, Eugene E., "A Machine-Independent Data Management System," Datamation (June, 1973).

Nolan, Richard L., "Computer Data Bases: The Future is Now," Harvard Business Review (Sept.-Oct., 1973).

Romberg, Bernhard W., "Data Bases: There Really is a Better Way to Manage Your Files," Infosystems (May, 1973).

Senko, M. E., Altman, E. B., Astrahan, M. M. and Fehder, P. L., "Data Structures and Accessing in Data-Base Systems," IBM Systems Journal (Volume twelve, Number One, 1973).

COMPUTER PERFORMANCE MEASUREMENT AND EVALUATION: A PERSPECTIVE

Tad B. Pinkerton
Associate Professor
Madison Academic Computing Center
University of Wisconsin

COMPUTER PERFORMANCE MEASUREMENT AND EVALUATION: A PERSPECTIVE

Performance measurement and evaluation has been a popular topic in the last few years. Cynics will say that this interest was motivated by the economic hard times which began several years ago. Whatever the initial motivation, there are many reasons for sustained interest in the subject. A primary reason is the fact that there can be a vast difference in performance between a poorly engineered program or system and a well-engineered one. Another reason is that we have learned that other people cannot be depended upon to create and deliver systems without major performance problems.

So we have strong interest in measurement and evaluation: consulting firms and hardware manufacturers and software houses who make most or all of their profits from the sale of measurement products and evaluation services. An ACM special interest group, SIGMETRICS, exists to focus work in the area. Several national conferences have been devoted exclusively to the subject - sessions on performance at other conferences are heavily attended.

An important aspect of the present concern is recognition of the concept of software engineering. A program or system is not solely a mathematical and logical algorithm, but something which operates in a time and space domain - the real world. Its performance, like that of an automobile or electronic circuit or any complicated piece of machinery, is often only established by measurement and improved by a feedback loop containing redesign, performance prediction, reimplementation and further measurement.

The popularity of measurement and evaluation today stems in part from its utility across the whole life cycle of a computer system: hardware design, hardware acquisition, subsequent reconfigurations, software design, subsequent software tuning, debugging and software maintenance, etc. It is also extremely valuable in computer science education and research. A detailed knowledge of what is really happening inside the computer is sometimes necessary in each of the above situations.

It is difficult to define precisely what we mean by an activity of such importance in so many areas of computer science. Indeed, the software engineering point of view says that we should not dignify it as a special activity at all - it should be completely integrated into all our other activities. In any case, we measure a variety of phenomena involving computers, including the behavior of hardware, operating system, language processors, applications programs, etc. But we also want to include the behavior of things in the larger man-machine system: polling delays in a communications line, and people at interactive terminals, for example. Evaluation of the measurement data is the second step in the process. We may obtain useful information by direct observation, but we must frequently analyze the relationships between data items in order to understand the dynamics of system variables. In abstracted form, the data is often useful in models.

Let's be more specific for awhile. The measurement process itself can be divided into four more or less distinct phases: collection, recording, reduction and analysis. At specific points in the time/space framework in which our system operates, events of interest occur: the scheduler starts a new job; at another time and place a program requests the expansion of its memory region. We can take two points of view - we can collect a datum or update a count each time that such an event takes place, or we can periodically sample the state of the system as it exists between the occurrences of these events. For example, we can save the size of a memory request each time a program expands, or we can occasionally sample the present size of the program.

Hardware and software monitoring are simply two different techniques for collecting event data or samples. In the first case we use physical probes to detect state changes in circuits. When a state change occurs, we can note the occurrence by incrementing a counter. The probing device must be able to detect the fastest state change and return (pulse width) that can occur, and also recover in time to detect the next one (repetition rate). In addition to counting once per state change or pulse, one often times the duration of an event by counting clock pulses. A typical monitor, of course, has many such probes active at once. Recording is typically done on magnetic tape, although an array of counters can be kept in memory for short experiments.

Software monitoring typically requires the addition of code to the software whose performance is to be measured, or to service routines called by that software. In addition to the obvious problem - that software measurement requires system resources and thus itself distorts the measured results - code implantation creates a maintenance problem when the software is supplied by someone else in the usual series of successive releases. Many software measurement products have avoided this problem by using a sampling process which periodically interrupts the subject program - from without, and snapshots the relevant state variables. Recording of data captured by software means also requires system resources, so that considerable care is required in its implementation.

Processing of performance data can be done in real-time to reduce the recording problem. However, this requires further system resources in the software case, and with hardware measurement there is usually not the time available for significant processing before the next event occurs. Thus we usually find that data is recorded for processing after the measurement experiment has been completed. Almost invariably, this takes two forms. First, it must be scaled, sorted into appropriate sequences, collated with other data, and selected by type and time interval. Sometimes the data must actually be corrected, for example, in synchronizing records time - stamped by two independent CPU clocks. We refer to this processing as data reduction - it is not unlike the initial processing of telemetry data from spacecraft or rocket systems.

Data analysis begins once the information has been processed to the point where individual items are significant in behavior or performance terms. It takes many forms: simply listing the data to form intuitive hypotheses, further processing and display to measure correlation and other relationships between variables, and abstraction and input to mathematical and simulation models.

Before we go further I must emphasize a very important activity - validation - which must pervade the measurement process. Familiar objects look very strange under a high-powered microscope. So it is with detailed performance data, and hence we have little redundancy and few reasonableness criteria with which to spot the inevitable errors. Anyone who has looked at the back-panel wiring on a computer of recent vintage can appreciate the probability of incorrect pin selection for probing. Probe placement in someone else's software and interpretation of the meaning of state variables are fraught with equal dangers. Many experiments are required to ensure that the words you collect mean exactly what you want them to mean!

What ~~advice can~~ we offer to the installation wanting to get started in the measurement business? Here are a few observations about what to expect:

1. Answering a particular question or solving a particular problem is an iterative process which converges when you have found the proper interpretation of the right data. Initial attempts always demonstrate that more data is needed and a new analysis must be performed.

2. The greatest success usually occurs when one uses several measurement and evaluation tools together. Validation and understanding of data is considerably enhanced by the stereoscopic effect of viewing an activity from more than one perspective. For example, the use of both hardware and software monitors on a problem yields independent and complementary data. Trace-driven modelling is another example: actual data from a system can be processed and used as input to a simulation model, instead of relying on approximate data sampled from more or less arbitrary distributions. At Wisconsin, we assess system changes with software monitor data from a set of benchmark programs. These programs were selected and combined using software monitoring to choose a representative workload.

3. Measurement and evaluation is most fruitfully treated as an *ongoing* process, as deserving of stability and maintenance as your most cherished application programs. The tools require time and effort to develop and understand. Such expertise leads to much faster solutions to specific problems.

Basic improvement of software performance can best be achieved by promoting and sustaining two attitudes on the part of software development staff:

- a measurement attitude

  No program should be designed and implemented in the first place without a plan to measure and understand its dynamics of operation. Means should exist during both testing and production use to monitor its performance.

- a performance model attitude

  No program should be designed and implemented in the first place without a rather detailed expectation (a performance model) of how it will behave. During implementation, testing and production, it should then be evaluated against that expectation. Modifications should be made first to the model and then to the program.

FUTURE TRENDS IN SYSTEM ARCHITECTURE

Arthur E. Scott
Assistant to the Director of
Application System Development
IBM Corporation

FUTURE TRENDS IN SYSTEM ARCHITECTURE

Arthur E. Scott

There is a fundamental set of human information processing tasks that can be supported by computer systems operating in the interactive mode. We acknowledge that sensor-based, batch, remote job entry, and teleprocessing types of systems will continue to exist. The area of concern of this presentation, however, was the burgeoning technological area called Information Systems. The major issues addressed were: (1) skill limitation (the fact that people with no professional programming skill will be the users of information systems), (2) modularity (the ability to grow in capability without major upheaval), (3) human usable systems (the ability to communicate with the system in the user's language), (4) system growth (where the goal is to implement from the easy-to-comprehend to the more sophisticated, and (5) strategy (a description of some "ultimate" goal and the logical and technological steps necessary to get from here to there).

The presentation is available under separate cover as a publication of the IBM Corporation. "Trends in System Architecture," Donald C. Burnstine and Robert B. Miller, A.R.-09 55-03. Copies are available by writing the Systems Development Division, Department B76, Exploratory Development, A. E. Scott, Manager.

FEATURE ANALYSIS

STUDENT FINANCIAL SYSTEMS

Dominic Bordonaro
Director
Systems & Data Processing
Ithaca College

R. Peter Jackson
Director
Student Records & Finance
Cornell University

Feature Analysis:

Student Financial System

R. Peter Jackson

Within the systems which support student services, one can identify three basic subsystem areas which serve Admissions, Registrar, Bursar and Financial Aid. Each is somewhat dependent on the other with the greatest dependence focusing on the subsystem which handles biographic and academic information - at times called Registrar or Student Records or Registration and Scheduling.

Today we will focus our attention on the student financial area which is dependent on the Registrar system, normally. In my frame of reference, the student financial area is composed of the activities which support most financial transactions with students.

Traditionally, the student financial arena include the Bursar's office and a Financial Aids office, which encompasses student loans and student employment, plus parts of other offices which process awards, or issue checks and the like. Historically, the two basic offices (Financial Aids and Bursar) were conceived and nurtured at opposite sides of the university house. The Bursar, of course, is identified with the business side along with accounting, auditing and budgeting, but financial aid usually evolved out of the dean of students domain, where counseling and advising have been paramount. The counseling orientation vs. the business orientation have separated these two families who basically serve one end - financial services to students.

Recent events have turned the two toward each other:

1. Federal and state monies going directly to students has become a major form of aid to higher education. This has resulted in greater exposure of these offices to more students. No longer are financial aid people understanding counselors with a little

money for those in need. In order to do their job they are becoming expert in parental income management, Federal and State funding, lobbying, and accounting. They must work with as many as five different needs analysis systems, plus their own, and still remain a bit of a "Mr. Chips". In order to do their job well they must try to make sense out of diverse funding patterns which have evolved with acronyms like EOP, EOG, HEOP, SEOP, HEAC, GILP, and most recent and descriptive - BOG. Yet, they must still keep in mind the main purpose of rendering students a broad range of financial assistance.

2. The bursar functions also have grown to reflect the complexity of financial transactions and within the area the pressure has been toward greater consolidation of institutional charges and credits in order to streamline billing function. Bursars are no longer issuing only tuition bills, but now the statements may include monthly rent for apartments, traffic fines and fees for adding and dropping courses.

The student and parent - college and university - State and Federal government is a complex financial network and has become more and more complicated. Clear, concise information and response is needed in order for colleges and universities to cope effectively with this burgeoning area.

The complexity of the financial picture for students has created a greater need for counseling and advising. Consequently, a greater demand exists for administrative information about the students' total financial circumstances and relationship with the university. Three examples: (1) In time of shrinking graduate school support, rules for fellowship sometimes restrict the recipient's ability to take on extra employment. Exception reporting from a payroll file which matches fellowship stipend with other income - assistantship, or simply part-time work enable the deans to monitor their programs more effectively and stretch available dollars further and more equitably, (2) New Federal programs, such as the Basic Opportunity Grant, are open to many but, at Cornell, we found only a few have responded; 300 students are qualified but 60 applied on their own without encouragement. Program directors need to be able to identify students from various income levels in order to identify those eligible for such support, (3) As pressure mounts for maximum use of all funds a need develops to insure that these funds receive maximum play - particularly the most restrictive. Thus, the requirement for matching the characteristics of the funds with the characteristics of the student. An example would be to link the student from Wayne County New York, majoring in Pomology, to the LaMont Scholarship restricted to pomology majors from Wayne County, New York.

These pressures for a complete student financial system should enable institutions to serve the student efficiently. But now, and of equal importance, good systems will help preserve and more adequately control institutional resources.

I would like to outline the functions which I see a student financial system serving, going through the basic steps as they occur in the annual cycles.

In the financial aid area there are two basic cycles; granting aid to new students and renewing aid for continuing students:

1. Identifying and Describing Financial Aid Candidates -

   The admission systems serve as the foundation of the financial aid process. Once the student is admitted and, if he applies for aid, his application is forwarded to the aid office for consideiating. Many functions in awarding aid to new students are duplicated in making upperclass awards, such as determining and assessing merit, checking for unique eligibility for restricted scholarship, etc.

2. Determination of Financial Need -

Need is considered one of the major criteria for aid in many institutions and many do not award beyond this level except in unusual circumstances. Need is now computed by several agencies, such as College Scholarship Service, American College Testing Service, the Federal Government, etc., plus whatever the university might do individually. Need computation is in a state of flux and although several CAUSE members have developed institutional programs which compute need, it might be best at this stage of evolution to capture what the service agencies develop and allow for simple internal adjustment.

3. Control of Financial Aid Funds -

In awarding aid to new students one must be able to operate on the basis of over-committing aid on a global basis to prospective student group because anywhere from 30% to 80% of those admitted eventually matriculate. This gamesmanship calls for close record keeping, taking into account history and experiences from past year, as well as sensing trends for the current year. Different types of awards experience different "takes"; for instance, National Scholarship which may be awarded on the basis of personal and academic qualifications will experience a different acceptance rate than a local scholarship which is awarded on the basis of home residence. Fund control is a necessity for the renewal of awards to continuing students as well as new students. In fund control, histories of the commitment as well as the expenditure are requisite plus adding weights for attrition rates, and changes in financial need, etc.

4. Packaging Awards -

The decision on the components of awards has been viewed as a personal "human" step in awarding aid - an art rather than a science. In reality this can be carried out mechanically once a "personal" decision has been made concerning the level of the award. Categories of awards, such as Good/Better/Best, or A-B-C classification can be given to candidates and then awards may be applied automatically. The classification could influence either percentage of scholarship assistance with the balance made up by a form of self-help, or the classification would influence the amount of self-help, such as level of $600, $900, or $1200.

Loans:

Long and short-term loans represent two different operational problems as short-term implies collection while in school and long-term implies collection upon leaving school.

Student Employment:

Part-time employment is an area coming back into the arena of respectability. The Federal College Work Study Program has given great emphasis to this form of self-help. To the systems person it presents a major interface problem - payroll and student systems as students are paid usually on the university payroll system, yet must be monitored and accounted for on the central Library.

5. Forecasting -

Linked to the award process forecasting the long-range impact of awards over a period of time to ascertain the institution's long-term aid commitment. Forecasting has two applications; forecasting the total impact for the year when packages are set at several levels and, secondly, projecting the total impact of the awards on the university budget in the current year and the next three or more years. Forecasting is called upon several times during the year and serves as a very valuable tool, particularly in this period of financial stringency.

6. Award Notification and the R.S.V.P. -

Telling students about their aid awards becomes a major task as it may involve 40% to 50% of all undergraduate and professional students and 90% of the graduate students. Automatic notification is an appropriate addition to financial service cycle. The process is an impersonal one now, with automatic typewriters, mimeo "fill-ins", etc., and direct automation would make no difference in this respect and would result in far fewer errors during the process.

7. Consolidated Billing -

Colleges and universities know that many offices rendered separate bills to students - an inefficient and ineffective practice. Billing can be reduced considerably and simplified by paring down to a single consolidated bill. The system would consolidate tuition, fees, dining and housing charges, music and physical education fees, health service charges, plus incorporate scholarship and loan credit, payments, corrections and, if germaine, even finance charges.

Within the billing process, institutions can make use of finance charges to regulate payments. Cornell operates on the basis that if tuition and other payments are not received within 50 days after the due date, finance charges are assessed at 1% per month. Although this may appear to be a loose arrangement, it gives the student an opportunity to set up a payment plan suitable to the circumstances. The finance charge counterbalances the cash flow problem and there is little need for individual student-staff administrative time and energy spent on setting up deferred payment plans and handling exceptions.

Another area deserving mention is a "charge card" arrangement whereby students charge bookstore items, meals, tickets, much like the Mastercharge, Bankamericard. Thus charges might be handled separately or be included in the main bill.

These two innovations are not without criticism practically and philosophically and deserve considerable discussion by the community before adopting.

8. Create Stipends -

As a conclusion to this student financial system stipends need to be created once all the credits exceed the debits and the student has money coming. Under certain circumstances stipends need to be issued on a monthly basis instead of a lump sum, and students should have the option for receiving money either way.

9. Reporting -

Reporting is different than forecasting. We all know the demands for reports on all levels are considerable - and growing. This demand will continue to grow as the long overdue cry for accountability rings out to higher education.

10. Organization of Financial Center -

One method of achieving a more complete and consolidated financial approach in the student area is to organize a financial center which brings together the Bursar and Financial Aids into one office, or two departments under one head. Within the financial service center one could identify three basic sections of activity:

1. Financial Counseling and Aid - The role of the traditional financial aids office.
2. Student Accounts and Billing - Role of the Bursar.
3. Financial Processing - The purpose of this activity would be to process charges and credits plus assemble all financial information related to the student. Information would come in from payroll, accounting, outside agencies, and internally in the consolidation of all financial information on students.

There are some dangers with this basic outline, particularly in the area of checks and balances. Please keep in mind that people involved are not natural bedfellows and one can expect more than one vigorous staff meeting should the two be drawn together.

The purpose of this type of organization and supporting systems should be designed to appear simple and human to students and to others who are served. However, it calls for complex systems interface, i.e., accounting, payroll, registrar, loan collection, admissions and, perhaps, even the alumni system. The objective is to make direct communication with the student as clear, complete and simple as possible. Considerable work behind the scenes is required to effect the appearance and relationship of simplicity.

The second purpose of consolidation is to respond to the strong force which is to develop a complete picture of students' financial status. To answer the question about (1) What is the complete financial contribution which the institution makes to the student, (2) How extensive is the Federal and State contribution to students, and (3) How will these factors ultimately influence the enrollment and sociological makeup of the institution.

Mr. Bordonaro - a realist - will discuss what has been done by members of CAUSE in the way of student financial systems and the more technical implication of integrated systems design necessary to a full "Student Financial Processing Center".

STUDENT FINANCIAL SYSTEMS FEATURE ANALYSIS: PART 2

Dominic Bordonaro
Director, Systems & Data Processing
Ithaca College

Mr. Jackson has presented us with a conceptual view of a student financial services center. The center contains a number of functions which have traditionally been identified as separate accounting functions at most institutions. Since these many individual systems have been functioning successfully for many years, one might ask - why consolidate or integrate them? Why not continue as in the past?

I would like to approach this with the premise that current financial pressures placed upon the student for his college education have resulted in the need for a greatly expanded number of financial assistance programs which the college must administer. This has brought about increased transaction volume and corresponding complexity to the information handling associated with these programs. To continue as we have in the past may not only result in a disservice to the student but to a less effective, less efficient financial administration program for the institution.

In order to meet the objectives of most data processing centers - to provide the institution's management with the necessary information on which to base decisions, and to provide the functional units of the institution with effective operational information handling systems - requires carefully planned, carefully designed information processing systems. Speed, efficiency, and accuracy nearly dictates systems consolidation, or at a minimum, the construction of simple systems interfaces, in order to achieve a bidirectional flow of data necessary to the recording and reporting requirements of the institution, government, or other agencies.

Whether to effect the integration of these functional student financial services through an organizational restructuring or through the information processing systems design, or through some combination of both, is an institutional alternative based upon local considerations. For purposes of our discussion, I would like to assume that some form of function consolidation is desirable and that an information handling system is required - one which would provide the necessary interfaces to such functional areas as admissions, current student data, the accounting control system, student loan records, student payrolls, student receivables, and of course, student financial aids history.

How do we solve the problem?
Where do we begin?
Software development is expensive.
Do we build it, borrow it, or buy it?

Of course, the answers to these questions may vary depending upon the institution's available resources and personal preferences, but I would urge you to consider the "borrow" option. To those who express the attitude that "no other institution can do it as well as we", I wish you well on your long, expensive development effort. To others, I would offer the CAUSE program library as a potential resource from which to obtain the desired programs, or at least a part of them.

In all probability, the programs obtained through CAUSE could not be used at your institution in their delivered state. Modifications are probably necessary to conform to local needs. The investment required to make the necessary modifications must be determined and carefully assessed before you finally decide to import a CAUSE system.

If a close inspection and evaluation of the CAUSE library holdings prove that the degree of difficulty and expense for modifications is too great, consider the value of the systems and programming documentation, and the techniques contained therein, as an aid toward the final "in-house" developmental effort. 20/20 hindsight is something we all have. The shorcomings of rejected systems designs can often prove to be very useful in efforts to develop a sound system, since you can apply needed improvements to the basic design approach rather than to reinvent the wheel. Even though you may have to rewrite programs to your own specifications, the design cycle time could still be reduced significantly.

A recent inspection of the CAUSE library index revealed eight contributed systems which treated various aspects of the student financial services area. I then asked for and subsequently received, the documentation for these systems. I would like to comment on what I found.

Four of the systems dealt solely with the student loan processing area, one with student accounts receivables; three with the broader area of student financial aids. You must understand that my investigation into these systems was solely on the basis of the documents received. Each system document varies widely in the degree which the system details are described. Due to the "sketchy" descriptions, in areas of these documents, my impressions toward certain aspects of the system may be somewhat inaccurate. Time did not permit a more aggressive investigation into the finer details

I would like to address my comments concerning these systems, starting with the Loan Processing category.

National Defense Loan Billing System - Library No. CSX 0023

This system consists of a set of 15 COBOL programs. It is currently running on an IBM 370/145 under OS/MVT. The master file is ISAM using social security number as its key. It is a stand-alone NDSL processing system. That is, all input, processing and output is confined to the

NDSL functions and does not interface with other processing systems. It provides a wide range of services necessary to the administration of the NDSL program. Current implementation is in a batch mode with a processing cycle of monthly.

The systems documentation infers that a loan master file is the only file maintained and therefore provides only current status information. Detail transactions are held for back-up. Account analysis must therefore be made on a basis of comparing one period file with another, or reconstruction of the detail against the master.

Student Loan Accounting System - Library No. CSX 0039

This system consists of a set of 19 COBOL programs plus several others written in PL/1. It is currently running on an IBM 370/155 in an OS/MVT-HASP environment. Facilities are provided for the processing of all types of loans, including NDSL. On-line capabilities are provided through program product - CICS. The office of student loans use IBM 2260 CRT terminals to maintain and query their data base in a real-time environment. A batch processing system provides monthly billings, fiscal year end updating, history file maintenance, and report generation.

This system could serve as a model to emphasize "user" responsibility toward the maintenance of the data base. It is responsive to the daily needs of loan administration. The documentation is not clear on one point concerning the handling of multiple loans to a single borrower. It appears that multiple bills are rendered for multiple loans and could cause some difficulty, or added manual labor, when a single payment check is received covering several loans. Distribution of principal and interest for each loan must then be computed manually.

Student Loan Accounting System - Library No. CSX 0040

Running on an IBM 360/50 under OS/MVT, this set of 19 COBOL programs is used to manage all types of loans, including NDSL. It is a sequentially organized file (SS number) and runs in a batch environment.

The file structure is an interesting one in that three segments are specified - one containing the borrower's biographic data (name, address, sex, age, etc.); a second segment describing the individual loans outstanding with this borrower, containing all data necessary to the management and collection of the loan; a third segment (attached to each loan segment) stores all processed transactions pertaining to the loan (payments, cancellations, adjustments, etc.). This last segment is purged each fiscal year, following the completion of annual reporting requirements. This technique permits the use of a single "master" file for all processing, reducing handling and maintenance tasks.

A consolidated bill is produced which reflects all due loans. Distribution of the payments applied against the various due loans must be performed by collection personnel.

This is a stand-alone system and does not tie directly to other student records or accounting systems, with the exception of one program which matches the NDSL master with the student data file while producing exit interview lists and mailing labels.

NOTE - Ithaca College, University of Southern California, and others, have adopted this system to their own needs without extreme difficulty.

Student Loan Accounting System - Library No. CSX 0041

This system consists of 14 COBOL plus 1 RPG program. It is running on a medium scale (65k) UNIVAC Series 70 computer in a DOS batch environment. CSX 0041 is an enhanced version of CSX 0040 (above) and is Ithaca College's recontribution to the CAUSE library. It retains the single master file philosophy and all processing functions of CSX 0040, and adds a number of desirable features, including: Data Collection - The student data file is attached, from which all biographic data is extracted to create the borrower's record, only after the student status has been verified. Loan dates and amounts are the only input needed from the loan office. All payment and cancellation transaction input is created automatically in the Comptroller's Office, through the use of Standard Register Source Record punches and is a by-product of their cash receipting operation. Addresses and other biographic data for all active students is automatically updated from the student data file record. Updates for non-active students are input by the Comptroller's Office via change sheets. Processing - An automatic distribution of payments program has been added to the processing stream which distributes all gross payments and cancellations, allocating principal and interest, across the multiple loans according to a set of rules established by the Comptroller. Output - In addition to certain report enhancements, the system automatically creates accounting distribution cards which, when fed into the general accounting system, charges or credits the proper accounts, relieving the Comptroller from this task while providing the proper audit trial and supporting documentation; loan disbursements result in the creation of card output which, when fed into the student accounts receivables system, will credit the student's account in the amount of the loan, or in the case of a loan reversal, will charge the student's account. Information Retrieval - The loan master file has been tied to a proprietary software system called EXTRACTO. This permits the Comptroller's personnel to write their own report requests which relieve our programming personnel for other tasks. A single passing of the loan file, against EXTRACTO request parameters, permit any number of reports to be generated. To date, the Comptroller's Office has amassed some 50-60 different report requests which they run at will, creating their own reporting system which is responsive to their own needs. The diversity of approach demonstrated in the four previously described loan processing systems, makes the selection of one, suitable to your needs, more realistic.

The next category of use deals with the student billing and collection process. I found one system in the library, as follows:

Student Accounts Receivable - Library No. CSX 0052

This system consists of 16 COBOL programs and is currently running on an IBM 360/50 under OS/MVT. It is a tape system with processing cycles established at weekly, monthly, end of term, and end of fiscal year intervals.

The student data file is used as the basis for establishing the accounts. The accounts receivable master is then fed with transactions resulting in billing or crediting of the student. This system handles all student charges and credits, except for student loans. Up to 99 different types of transactions are permitted in the coding structure. One of the interesting aspects of this system is that it handles both installment and non-installment contract charges, and all agency records as well. It approaches what could be considered to be a true consolidated billing system.

Its straight forward approach would permit easy modification to accommodate local needs.

The final category of systems listed with CAUSE involve student financial aids processing. These system descriptions more closely approximate the service needs for the student financial services center presented a little earlier by Mr. Jackson. Let me briefly present them and then I would like to comment on a set of desirable systems functional characteristics necessary to the "Service Center" concept.

Student Financial Aid Grant and Loan Awarding System - Library No. CSX 0036

Only six COBOL programs are involved in this system. It is currently implemented on an IBM 360/40 under DOS. All processing is handled via tape.

This write-up suggests that these six programs are only a part of the financial aids processing system. It states "This is a sub-system of the university's financial aid system which maintains and processes total data concerning student financial needs and all available funds at the university. It creates and updates a financial need file from student or parent confidential financial statements, matching this file with the admissions file and current registration file creates a scholarship record in the scholarship file. Based on given parameters, awards are created for merit grants, long term loans, NDSL and EOG. Corrections are made and letters are printed for all students that have applied."

Payroll data is fed into this system to provide a more complete picture of all financial assistance. A history record details previous years awards. One interesting feature is that a memo slip is created for awards, payable to the student, valid only at the university Bursar's office. One could just as easily apply this credit directly to the student accounts receivable file, if local policy permits.

This system has a very simple structure, but yet satisfies a number of important needs.

Student Financial Aids Data Processing Services System - Library No. CSX 0067

This system asbstact is a little more difficult to assess because it describes sub-components of a broad data base systems approach in which the student financial aids functions are imbedded. No separate information is listed regarding the number of programs involved with these services since a number of them, particularly the edit and maintenance functions, are really a part of the overall student records management system. The hardware involves an IBM 360/40 running under DOS and includes teleprocessing facilities (2260/2740). It is a little unclear to me just how on-line terminals are used here since the system flow chart identifies only on-line data collection with a subsequent batch updating of the files involved. Whether or not inquiry capabilities are included cannot be determined.

The writeup describes briefly all of the financial assistance programs normally managed by a financial aids center, including loans and payrolls - both work study and non-work study. I assume that most of these functions are included, but are not described individually. Although the data pertaining to these assistance programs is captured by the system, I sense that most of the computation and decision making occurs at the financial aids office prior to input.

I find that the approach used with this system offers a great potential toward the achievement of the ever so illusive totally integrated system - MIS. Closer inspection of this system is strongly advised.

Student Financial Aids System - Library No. CSX 0079

This system writeup does not list the number of programs involved, but the COBOL language is stated. It currently operates on an IBM 370/65 under O/S MVT-HASP. The system maintains a financial aids tape file which is created by input from the financial aid office and the registrar's address file (also tape).

This system creates a number of reports, analyses, labels, cards, award letters, etc., and is integrated with the payroll system. It creates budgetary credits for accounting as well as output to the loan system files. I could not determine whether the programs monitored the financial aid applicant's qualifications or determined the need levels, but reports containing this resultant information are produced.

The documentation for this system is rather detailed - too much so, for this appraisal. It contains all information except for the program source code, including the initial systems study, clerical procedures, agreements, etc. It appears to use a very basic approach, with numerous individual job steps. The copy quality of this document was poor and difficult to read - a source of confusion to me. This should not discourage anyone from looking further into its use. The financial aids office requirements do appear to be met.

I have not attempted here to identify a system which is the "best" solution to the problem. Institutional differences would not permit it. I would only suggest that the problem you are trying to solve might be solved, at least in part, through a brief inspection of the solutions presented by others. CAUSE may be of some assistance to each of us in this regard.

In conclusion, I would like to list a set of desirable functional characteristics which I feel are necessary to the student financial services center concept. It should:

- Determine the students qualifications for receiving financial aid.
- Determine level of financial need.
- Permit optional alteration of the computed financial need figures.
- Provide facilities for the handling of a broad range of assistance programs, such as EOG, EOP, WSP, NDSL, BOG, others.
- Permit a set of variable parameters controlling award "packaging".
- Incorporate a facility to simulate the awards disbursements process - (trial run).
- Include a funds control mechanism.
- Prepare actual awards disbursements, once acceptable "packaging" has been determined.
- Prepare award notification letters to each F/A applicant.
- Permit automatic award reversals - forfeit or cancellations.
- Produce reports for exceptions and/or erroneous conditions.
- Prepare a wide range of up to date operational reports and statistics.
- Permit a facility to produce simple "one time" informational reports or response to inquiries.
- Maintain a history of all student financial aid transactions.

The system should be integrated with or provide convenient interfaces to:

- Admissions data.
- Current student data.
- Accounting system data.
- Consolidated student A/R system.
- Loan data.
- Student payroll data.

Finally, the system should not only provide financial analysis of past financial assistance programs history, but should provide assistance with the task of projecting the financial aids funding needs of the institution.

KEYNOTE ADDRESS

APL AND ITS USE IN EDUCATION

Kenneth E. Iverson
IBM Fellow
Scientific Center
Philadelphia, Pennsylvania

KEYNOTE ADDRESS

APL AND ITS USE IN EDUCATION

APL is a simple and powerful general-purpose programming language, available in a number of interactive terminal-based implementations. The speaker discussed the salient characteristics of the language, the several modes of use on the language in teaching (direct, CAI, and simulation) and presented usage within universities in the United States and abroad.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

INTER-INSTITUTIONAL SYSTEMS DEVELOPMENT

Jack W. Smith
Associate Director
Management Information Systems
State University System of Florida

INTERINSTITUTIONAL SYSTEMS DEVELOPMENT

The development of information systems to support multiple institutions of higher education is a popular topic of conversation at this time. Systems of educational institutions as well as voluntary consortia are organizing in an attempt to address common problems and share resources. The need for comparable data to answer the better questions being asked by those who are responsible for funding higher education is the major reason for these efforts. Secondarily, those responsible for central Information Systems are realizing that a better position for attacking their problems is through the development of sound systems at the operational level in concert with specification of needs at the higher levels of administration.

From the experiences gained in this arena in Florida, three (3) major areas of concern contributed to both the successes and failures:

1. The active involvement of the user personnel throughout the entire systems development cycle.
2. The existence and enforcement of programming and documentation standards.
3. The ability to intelligently determine the information impact of the new system on the campus in light of its existing systems.

Using good backward vision generated by the installation of the accounting system in six (6) institutions, the admissions system in seven (7) institutions and participating in other projects where systems are shared even across vendor hardware and across severe software gaps within a vendor, some techniques for addressing these points have been developed. These will be discussed individually under the titles of:

1. User Role in the Design and Development of Systems -
   Mr. Olcie Harrell
   Associate Director
   Business Services
   State University Systems of Florida

2. Systems Analyst Point of View of Computer Systems Design With Sharing in Mind -
   Mr. Earl Kellow
   Systems Consultant
   UNIFTRAN
   State University System of Florida

3. A Technique for Systematically Evaluating the Information Impact of a New System -
   Mr. Dennis Klinger
   Associate Director
   Southeast Regional Data Center
   Florida International University

USERS ROLE IN THE DESIGN AND DEVELOPMENT OF SYSTEMS

The State University System of Florida began the design and development of one system in 1969 to handle all the basic accounting transactions needed for departmental ledger and a fund control general ledger. This system now consists of forty-one programs and was entitled CORE because we later identified seven other operating activities for which systems were needed to feed the CORE System or from which data could be extracted.

The primary and foremost concept desired in the new systems effort was to have UNIFORM TRANSACTIONS that could be transferred between systems that would produce common information that could be exchanged and/or compared with other universities within our state.

The CORE System was installed at three institutions in July 1971. Several problems became apparent rather quickly. From a retrospective view, the problems were caused by the following reasons:

1. Insufficient cooperation between the users of the system and the systems analyst;
2. Poorly described objectives by the users as to the controls required, printouts and formats desired, and multiple sort sequences of data beneficial to the users;
3. Inadequate pilot testing of the NEW programs via the use of cross-over tables;
4. Programming errors;
5. No training aids;
6. Inadequate operating manuals or documentation.

It is needless to say that the first year of operation using the system was rough; however, benefits accrue from mistakes and seven other systems, which will be explained shortly, started at a later date were the benefactors. Today, the State University System of Florida has six institutions and the Board of Regents using this one common system.

The seven other operating activities which were identified for systems were:

1. Personnel/Payroll,
2. Purchasing,
3. Student Financial Aid,
4. Contracts and Grants,
5. Accounts Receivable,
6. Property,
7. Local Funds.

The primary concept of UNIFORM TRANSACTIONS was to be maintained throughout the design of all the systems whereby all the systems would interface to use the basic transaction regardless of the initial source of input. The design and development of these seven systems began in 1971; however, to benefit from prior mistakes a method was instituted to control the development of these systems.

The method used was to divide the development of a system into three steps called LEVELS. The three levels were:

1. Problem Definitions,
2. Design,
3. Focus and Implementation.

Each level is described on separately attached sheets.

The personnel resources to develop seven systems simultaneously was recognized as being horrendous. Therefore, the alternative selected was for each university within the state to develop one system and then provide the completed system to all the other universities. This approach was called the HOST INSTITUTION CONCEPT.

Each host institution developing a system used a TASK FORCE COMMITTEE to coordinate the design, development and implementation of one system. The Task Force Committee was composed of both system users and analysts. The committee members consisted of representatives of one or multiple universities.

The maintenance for the completed systems is controlled via the same review process as the development levels, except that the completed systems are moved to one central office, The Board of Regents, for centralized coordination.

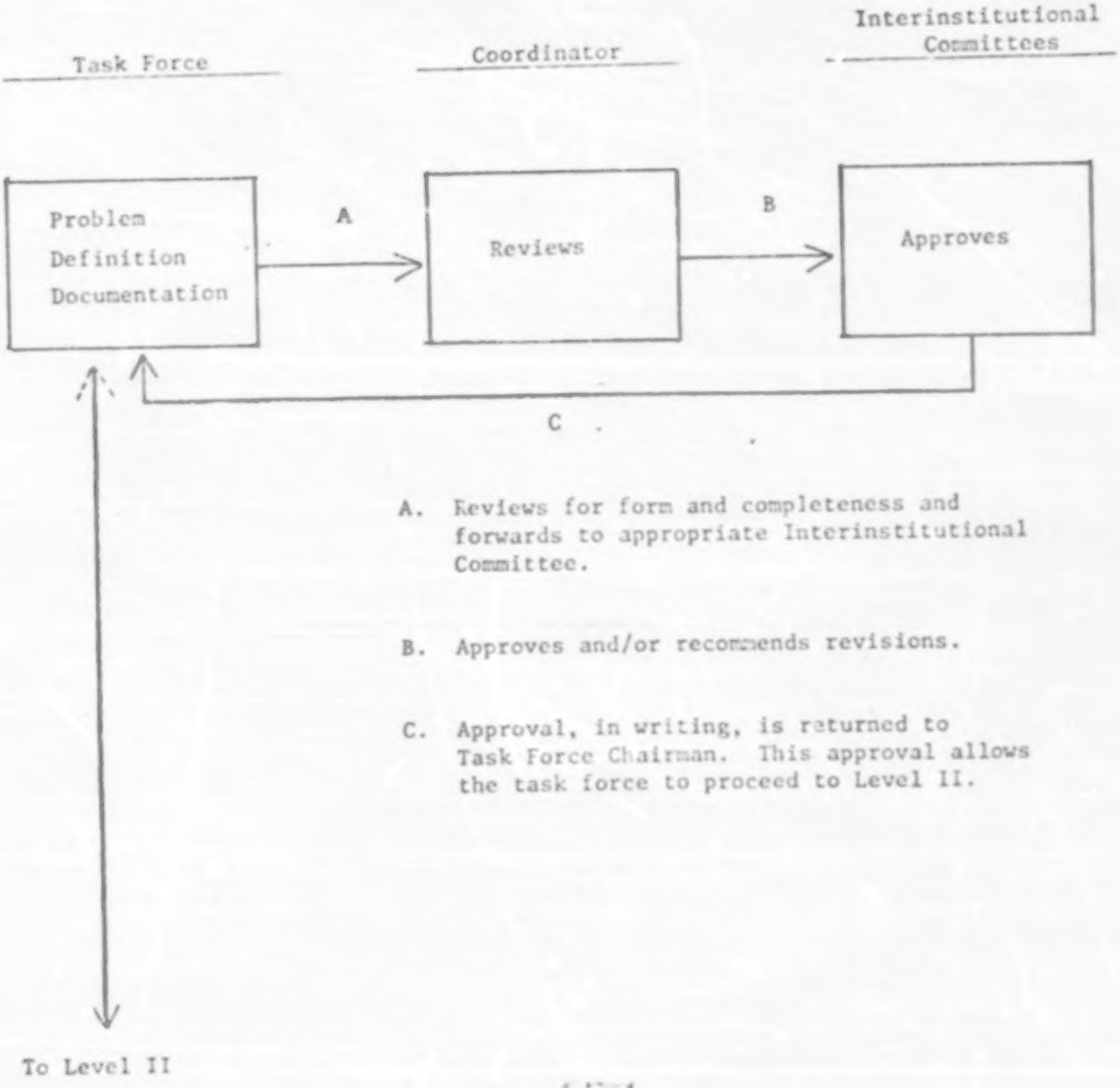
STATE UNIVERSITY SYSTEM OF FLORIDA
PRESENTS
UNIFTRAN

LEVEL I DESIGN

Problem Definition

Documentation which contains the following minimum components:

1. Objectives
2. Features
3. Interface Identification
4. Data Elements


Task Force
Coordinator
Interinstitutional Committees
Problem Definition Documentation
A
Reviews
B
Approves
C
To Level II

A. Reviews for form and completeness and forwards to appropriate Interinstitutional Committee.

B. Approves and/or recommends revisions.

C. Approval, in writing, is returned to Task Force Chairman. This approval allows the task force to proceed to Level II.

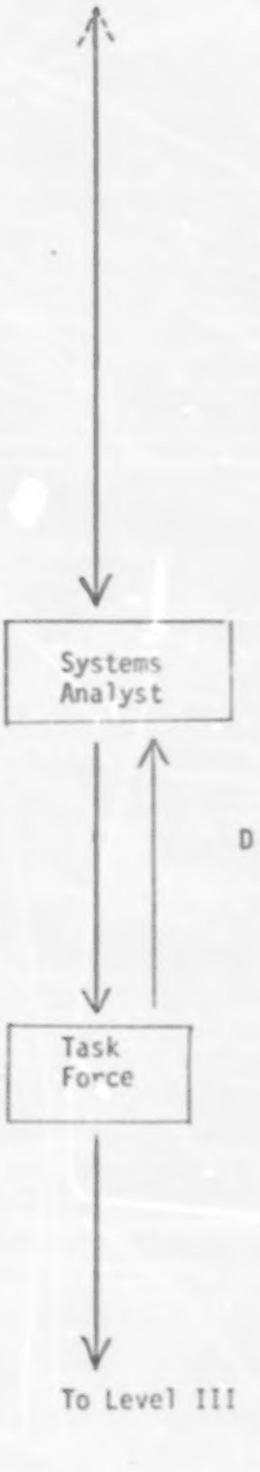
STATE UNIVERSITY SYSTEM OF FLORIDA
PRESENTS
UNIFTRAN

LEVEL II DESIGN

Analysis

The process of refining the document completed in Level I. This process should produce the following minimum additional components:

1. Analysis of old system
2. Flowcharts
3. Outputs defined
4. Inputs defined



The interaction between the systems analyst and the task force should result in the refinement of the Level I document and inclusion of the additional components listed above.

Any problems identified at this level should be resolved at this level. Interinstitutional Committee action may be needed.

Any change from original objectives should be approved by Interinstitutional Committees.

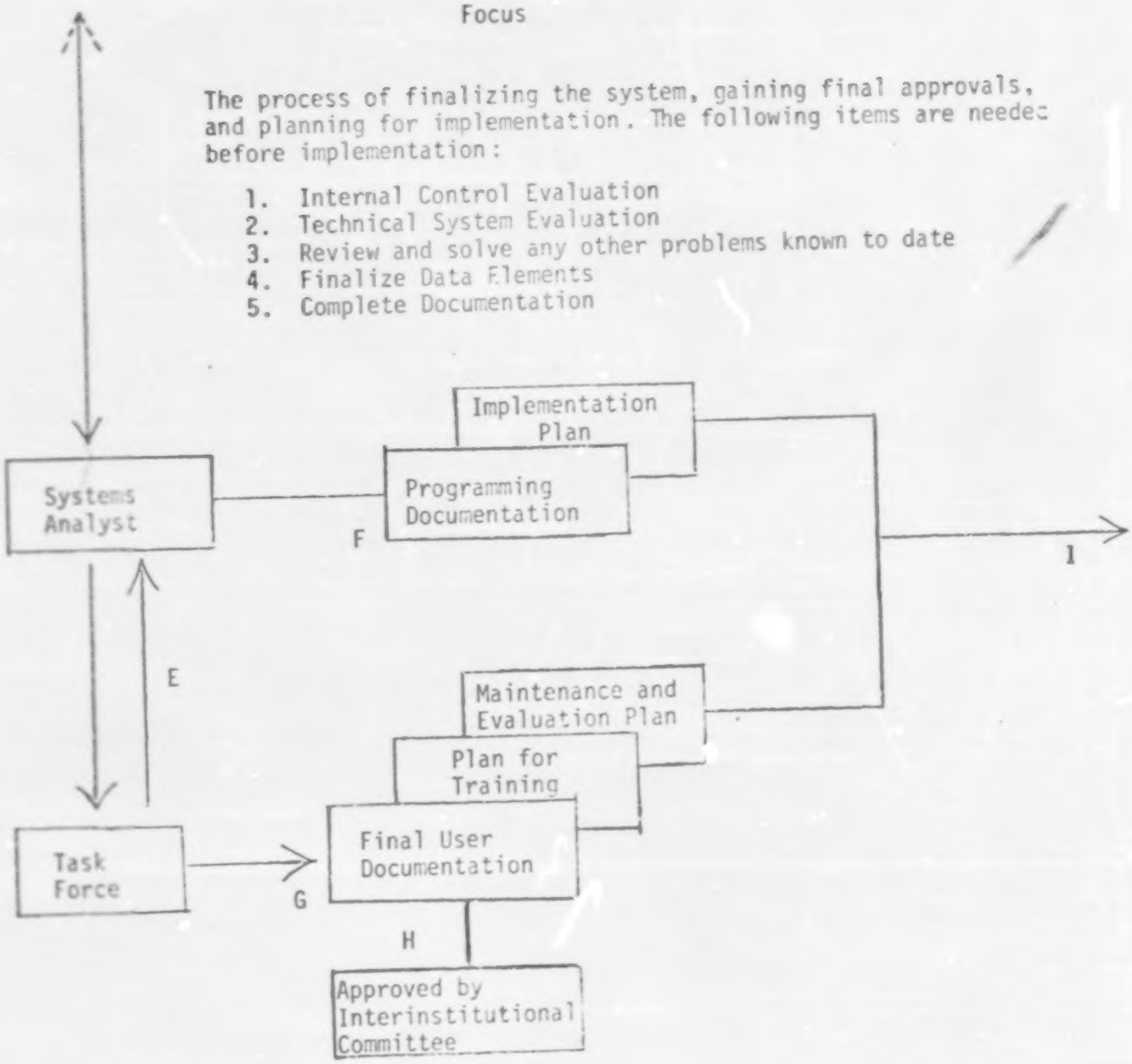
STATE UNIVERSITY SYSTEM OF FLORIDA
PRESENTS
UNIFTRAN

LEVEL III DESIGN

Focus

The process of finalizing the system, gaining final approvals, and planning for implementation. The following items are needed before implementation:

1. Internal Control Evaluation
2. Technical System Evaluation
3. Review and solve any other problems known to date
4. Finalize Data Elements
5. Complete Documentation


Implementation Plan
Systems Analyst
Programming Documentation
F
I
E
Maintenance and Evaluation Plan
Plan for Training
Task Force
Final User Documentation
G
H
Approved by Interinstitutional Committee

E Resolves final problems between users and systems people.

F Provides technical document for specification and programming.

G Provides final user document to be used as a basis for interinstitutional user manuals.

Provides a recommendation or plan to achieve training prior to implementation.

H Final approval in writing from the appropriate level committee

I Implementation.

SYSTEMS ANALYST POINT OF VIEW OF COMPUTER SYSTEMS DESIGN WITH SHARING IN MIND

Earl Kellow
Systems Consultant
Board of Regents

The keyword in the title of this address is "sharing". As a systems analyst what are some of the things which may take a higher priority when we are doing a system that we know, from the start, will be shared. This differs from a system that is already operating at one facility and another facility picks it up for their own use. Here, we are going to deal with the concept of a nonexistant system which, when completed will be installed in multiple facilities.

I would like to limit somewhat the scope of this presentation by dealing mostly with the programming considerations of a shared system. I believe there are a good many things which the analyst can do during the design of files, programs and general concepts that when implemented in the coding phase, leads to a smoother transfer of the system between facilities. Also, once it is shared at various facilities, it can be easily maintained.

I sincerely believe these two points (transferability and maintainability) are extremely important when building a set of programs for sharing. I'm making a basic assumption here that the system works, it does what it's supposed to do. Again, the main questions are how smoothly can the system be transferred from one site to another and how easily can it be maintained once it is operational.

Now, how do we go about making a package of programs that not only work, but we as computer people can easily work with.

As a case in point I would like to use the UNIFTRAN CORE Financial System of the State University System of Florida. I have been involved with the development and implementation of this system for almost four (4) years now. We have installed it in six (6) of the nine (9) state universities. One more will be coming on board July first 1974 and the other two are currently studying possible implementation procedures. It is also installed in the Board of Regents Office. This system is the general accounting subset with the end product being a budgetary and General ledger for each University.

The CORE Financial System is just one (1) of eight (8) systems which make up the UNIFTRAN package of the State University System of Florida. But it is unique in that it was the first one to go operational and I believe we learned some good lessons from it.

As far as the environment is concerned we have nine (9) state universities in our system. Enrollment is anywhere from three thousand

(3,000) to twenty-five thousand (25,000). During the brief life of CORE it has had to run it on an array of different computers. Honeywell 1200, IBM-360, Model 40 DOS, IBM 370-165, and UNIVAC 1106. So you can see we have small computers, large computers and several different manufacturers involved.

The State University System is also heavily involved in the regional data center concept. So we have the CORE financial system being run from remote terminals. It's probably best to sum up our environment as being - mixed.

Before we started coding any of the UNIFTRAN systems we had to lay down some ground rules. We decided that we were dealing with generalized systems. They were created to solve the major operational problems of all nine universities. A university would not necessarily use everything built into each system. There would be unique and special considerations at some universities that a system could not address.

Still, we wanted to provide as much flexibility within the basic framework as we possibly could. One way to do this is to limit as much as possible hard coded tables and lists in the programs. In the case of the CORE financial system we didn't want to have to go in and change the programs everytime an accountant wanted to add a department number or change a name on his chart-of-accounts. Another thing we did to help promote flexibility was to limit the number of tasks any given program had to accomplish. Making smaller more limited programs allows a university to build or tailor the system more to its own needs by picking and choosing among them. We realized we were increasing the number of individual programs that would make up the system. But we still thought it would be easier to handle. So again, we were concerned with generalized programs and promoting flexibility.

We then considered size constraints. The size of a program and the amount of resources it demands at run time, impacts the transferability of the system itself. We established the minimum configuration needed to run any UNIFTRAN system would be: one hundred (100) K of CORE storage, four (4) tape drives, and at least one (1) direct-access device (disk or drum). Our objective in setting the minimum configuration as we did was to allow every university in the state to run any UNIFTRAN system. I stated previously we would limit the number of tasks of the programs. This helps to get away from any monster programs that tend to eat up tremendous amounts of CORE storage and are difficult to understand.

File size and access methods also have a bearing on the transferability and maintenance of the system. In order to help compatibility the record length on any Indexed-sequential file must be a multiple of six (6). This meets a UNIVAC Indexed-sequential length requirement, Only fixed-length records can be used, No variable-length records and no packed decimal format can be used.

The problem of passing external data such as switches and overriding the current date had to be resolved. In DOS we had the UPSI Switch. For OS we had the PARM field and for UNIVAC and Honeywell we developed our own COMREG MACRO. Even though the switches were passed differently, depending on the operating system, we standardized the settings. The only two possible values were zero (0) for off and one (1) for on. These restrictions help to guarantee transferability of the system between manufacturers, And I don't feel they hinder the ability of the system to do its job.

At this point we have defined our minimum configuration. We have set up our size constraints and our file and access requirements. Now we need to turn our attention to the actual coding of the programs. What can be done during this phase to help make the system transferable and maintainable.

The coding language was established as ANSI COBOL. Now, this was a limited version of ANSI COBOL; not the full version. To try and maximize compatibility between manufacturers we did not allow the use of Report Writer, Internal Sort, and the table indexing features of ANSI COBOL. The intent here was to try and use a basic COBOL subset avoiding as much as possible the gray areas of COBOL which may tend to tie the programs to one manufacturer. A set of coding standards was developed to try and avoid possible inconsistencies between manufacturers. Beyond that a coding philosophy was developed which we try to ingrain in the training of our programmers. This philosophy is built around the old "Kiss" method, or Keep It Simple and Straight forward. We are not trying to build programs that are extremely tight and efficient which only one programmer can understand. Where one position of CORE is used fifteen different ways depending on where in the logic you happen to be looking. And where it takes another programmer three (3) months to figure out the logic paths enough where he can make minor changes to it without fear of bombing the whole thing. That's not what we want. We are seeking a simple and straight forward program that works. As the programmer is doing the coding he must keep in mind that at least nine (9) other programmers (one for each university) will be looking at his program to install it and maintain it. Also the man who actually coded the program will probably not be assigned to maintain it.

In order to help guide the programmer we have laid down coding standards for such things as; use of level numbers, performs, paragraph names, indention of if statements, one instruction per line, many note statements, etc.

Even with the coding standards and restrictions a programmer must work under, we do not feel we are hampering his basic ability to be creative. For each programmer has his own unique style and that style shows through. So, we hope the finished product will be a program that is simply written, uses straightforward logic, and can be quickly and easily understood by any competent programmer. Now that's a large order but it can be done.

The CORE financial system has proven that these concepts work. In 1970 the system was coded, compiled, and tested on an IBM 360-65 under OS with HASP. Since then we have converted the programs to run on a Honeywell 1200, UNIVAC 1106, and other Model IBM machines. The changes needed were relatively slight. Mostly in the Environment Division with the Select and Assign statements. The Procedure Division required very little change. I believe it was mostly having to do with minor differences in the Indexed-sequential reads and writes between manufacturers. So, we have established that the system can be easily transferred.

As for the maintenance question: the system has undergone a good deal of fine tuning and additions to it. The changes are made at each university with little problem.

In closing to reiterate my original theme, when you are developing systems for sharing, the questions of transferability and maintainability must assume a higher priority than normal. When you think that the programs that make up a system will not be installed just once but many times, and many programmers will be trying to understand them, then it makes sense to take a little extra time in the coding phase to help out later. This extra time taken will be returned many times over for each facility the system is installed in.

A TECHNIQUE FOR SYSTEMATICALLY EVALUATING THE INFORMATION IMPACT OF A NEW SYSTEM

A prototype Information Management System (IMS) is under development to provide the Southeast Regional Data Center (SERDAC) with the capability to manage the data and information associated with the various academic and administrative applications at Florida International University (FIU). As a long run goal, IMS will develop the foundation of data and information control necessary for SERDAC to implement an organizational data base to support FIU's applications. Without this management capability successful implementation of an organizational data base is not possible.

The management of data and information with IMS consists of three main activities and their supporting documentation:

1. Determining ambiguity in the use of data through a Data Inventory,
2. Identifying duplication in reporting information through an Information Inventory, and
3. Reviewing user information needs through both Inventories.

IMS flow diagram

A flow diagram of IMS appears in Exhibit 1 following the narrative part of the Overview. The three main activities listed above are shown across the middle of the diagram with the Data Inventory and Information Inventory supporting documentation feeding into them. To perform these activities a Data Administrator position needs to be added to the SERDAC staff. The person in this position would generate specifications for (a) improving system efficiency by eliminating ambiguity and removing duplication and (b) improving system effectiveness by responding more closely to user needs.

Neglected resources

Data and information are two of the most neglected resources in many organizations. Poor management of these resources significantly lowers the quality of the organization's decision making ability. Improved data and information management provide greater assurance that inputs to the decision processes will be acceptable. This in turn improves the quality of the decisions made at all levels in the organization. The work done to date on IMS represents a beginning in the improved management of the data and information resources of FIU. The application of this approach would provide the same benefits for other organizations in the State University System.

IMS Activities

Determining ambiguity

Many data elements are common to the various programs of the applications in a data processing system. (As used here data element means the name, usually appearing as a column heading on a report, which identifies a set of data values.) Since applications are developed over time by different individuals, numerous situations arise where common data elements are not referenced in the same manner. As a result ambiguity in the meaning of data elements gradually creeps into applications. Two types of ambiguity occur in these situations.

Redundancy.--First is redundancy where the same data element is referenced by more than one name. Sometimes these synonyms are justified and other times not. In cases where the redundancy is acceptable, a means should be available to document accepted data element synonyms. In other cases where it cannot be justified, a means should be available to identify and remove redundancy from applications. This will help insure a common understanding of the meaning of data elements.

Inconsistency.--The second kind of ambiguity, inconsistency, also creeps into the development of applications over time. In this case the same data element name acquires more than one meaning. This absence of uniqueness in data element meaning leads to confusion both in the minds of systems personnel as well as users. When the inconsistent data element name is used there is no unambigous way for all concerned to insure which of two or more meanings is intended in a given situation. This type of ambiguity should be identified and eliminated.

The Data Inventory of IMS provides a mechanism for controlling ambiguity in the use of data elements. This inventory consists of three forms of documentation indicated as input on the left side of Exhibit 1:

1. Term Directory (Exhibit 2)
2. Data Index (Exhibit 3)
3. Data Dictionary/Directory (Exhibit 4)

Each of these forms of documentation is briefly discussed in the IMS Documentation section of this Overview.

Identifying duplication

Some of the outputs of the various applications contain the same information. In some cases this duplication of information is necessary. For example the same information may be presented in conjunction with another group for a different user. In other cases where applications are developed over time by different individuals, duplication will creep into the outputs simply because personnel are not aware of the availability of the information in existing outputs.

In situations where duplication of information does not have a rationale, it should be identified and eliminated. However installations do not have a means for systematically documenting the current outputs of their various applications. A capability is needed to carefully identify the areas of duplication so they can be considered for elimination.

The Information Inventory of IMS provides the mechanism for performing this activity. This Inventory consists of four types of documentation indicated as input on the right side of Exhibit 1:

1. Report Index (Exhibit 5)

2. Report Copy

3. Report Profile

4. Information Directory

Each of these forms of documentation is briefly discussed in the IMS Documentation section of this Overview.

Reviewing user needs

The above two activities of IMS involve continuous monitoring by the Data Administrator of the various applications to determine ambiguity and to identify duplication in order to initiate the system improvements indicated at the bottom of Exhibit 1. The third activity focuses on the interface between SERDAC staff and user groups in reviewing information needs to improve outputs. Two of the necessary prerequisites to the successful completion of this activity are supported by IMS.

Data meanings.--First in order to respond to user requirements, the SERDAC staff and the user need to clearly understand what kinds of data are being considered. The availability of the Data Inventory meets this need. The Data Dictionary/Directory (DDD) provides clear and unambiguous specification of the meaning and use of each data element. With this documentation both the user and the SERDAC staff know exactly what is meant in a discussion of needs with reference to a particular data elements.

Information availability.--Second the staff and users need to know what information is currently available as output. It may be that given requirements of the user can be met with existing outputs. The Information Inventory provides a complete alphabetical listing of all information available on the output reports of current applications. With this documentation the staff and user personnel can quickly check the listing to see if desired information is currently available.

In addition to these two necessary prerequisites for successful response to user needs, the Data Dictionary/Directory and Information Directory may be used to assist the staff and user in identifying new needs. These two forms of documentation may be viewed respectively as the SERDAC data "menu" and information "menu" which may be used to suggest the type of data or information which will best fit the needs of the user. In this way the documentation indicates outputs which can be considered to respond to the problem that the user has defined. Taken together these aspects of reviewing user needs lead to the system improvements indicated at the bottom of Exhibit 1.

IMS Documentation

The Data Inventory

As previously listed there are three forms of documentation in the IMS Data Inventory. Examples of this documentation are attached.

Term Directory.--Exhibit 2 shows the first page of the Term Directory. This provides a list of all single words that are used to form the data element names in the Data Dictionary/Directory. For each term the standard abbreviation is indicated. The standard data element names are formed from the terms, and the standard program variable names are formed from the abbreviations. Using the Term Directory insures that a particular term will be used in exactly the same way regardless of the data element name that it appears in.

Data Index.--Exhibit 3 represents the first page of the Data Index. This documentation contains a list of all data elements in the Data Dictionary/Directory along with the stem for the program (COBOL) variable names and the code which is used to locate the record entry in the DDD file. This code is the key for accessing the entry from the file for the indicated data element. These codes are assigned to maintain the Data Index in alphabetical order. Sorting the codes into alphabetical order insures that the corresponding data elements are also in alphabetical order.

Data Dictionary/Directory.--The DDD entry for the data element "administrative function" appears in Exhibit 4. In its current implementation the DDD contains nine major fields in the data element record entry. These are identified in the listing for the entry by the major outline headings A through I. For example the first major field A contains the code for the data element which appears on the Data Index in the previous Exhibit. In addition to the definition for the data element which appears in field F, the entry also includes the standard set of data values for the element in field H. In this way the DDD establishes standards for the data values as well as the corresponding data elements.

The Information Inventory

Four forms of documentation for the Information Inventory are listed previously in the Overview.

Report Index.--Exhibit 5 is the first page of the Report Index which lists all output reports in the Information Inventory along with their unique identifying code in the first column and the standard SERDAC identifier in the last column. This list is used to obtain the code for a particular report given its name.

Report Copy.--This part of the documentation consists of notebook binders containing selected pages for all reports in the Information Inventory. This provides a ready reference for the format of the major academic and administrative reports.

Report Profile.--Since reports from different systems do not use standard terminology to refer to their contents, it is necessary to convert the data elements of the reports to the standard established by the DDD. In addition to standardizing the terminology it is necessary to represent the content of all reports in the same format in order to precisely compare them to identify duplicate information.

Information Directory.--In order to arrive at the Information Directory it is necessary to bring together in a single list all the information elements contained on each report in the Information Inventory. This is accomplished by sorting each information element from each report (e.g. 35 information elements on DLSU) to group all like information elements together in the Information Directory.

Summary

When fully implemented the Data Inventory and Information Inventory documentation in IMS will provide SERDAC with a mechanism for managing the data and information in FIU's academic and administrative applications. As shown in Exhibit 1, the information management function consists of three main activities: determining data ambiguity, identifying information duplication, and reviewing user needs.

A Data Administrator needs to be added to the SERDAC staff to perform these activities which are not adequately covered by the responsibilities of present staff positions. IMS and the Data Administrator will provide SERDAC with the resources necessary to eventually implement an organizational data base for FIU. Without these resources such an undertaking cannot be accomplished. In the meantime these same activities will improve both the efficiency and effectiveness of existing systems.

SOUTHEAST REGIONAL DATA CENTER

Florida International University

Information Management System

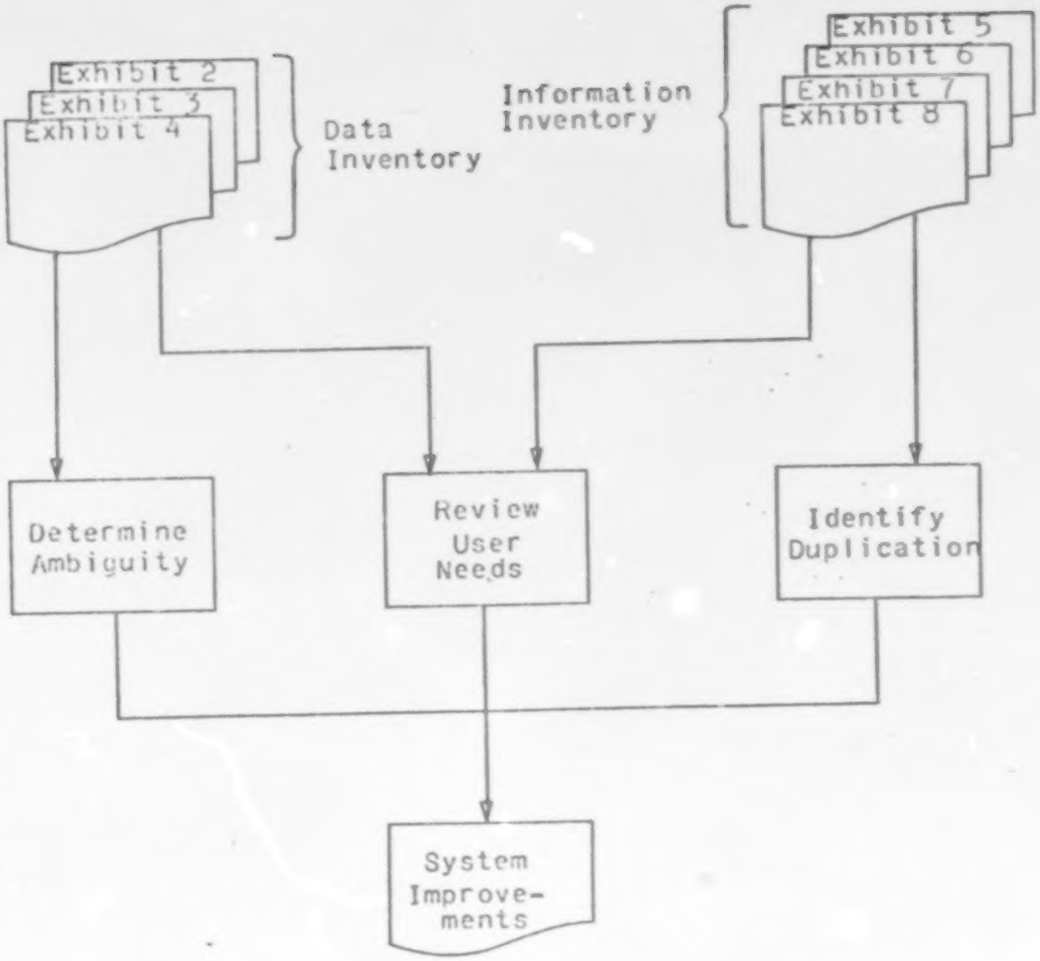


Exhibit 2
Exhibit 3
Exhibit 4
Data Inventory
Information Inventory
Exhibit 5
Exhibit 6
Exhibit 7
Exhibit 8
Determine Ambiguity
Review User Needs
Identify Duplication
System Improvements

Exhibit 1

SOUTHEAST REGIONAL DATA CENTER
Florida International University
IMS Data Inventory

Term Directory

Term	Abbreviation
Abbreviation	ABBR
Academic	ACDM
Account	ACCT
Audit	AUD
Acquisition	ACQ
Action	ACT
Activity	ACTV
Actual	ACTL
Administrative	ADMN
Admission	ADM
Aged	AGED
Agency	AGCY
Aid	AID
Alternate	ALT
Amount	AMT
Anniversary	ANV
Annual	ANL
Application	APP
Applied	APPD
Area	AREA
Attendance	ATTND
Available	AVL
Balance	BAL
Birth	BIRTH
Biweekly	BIW
Block	BLOCK
Book	BOOK
Budget	BUDG
Building	BLDG
Card	CARD
Cash	CASH
Category	CATG

Exhibit 2

SOUTHEAST REGIONAL DATA CENTER

Florida International University

IMS Data Inventory

Data Index

Code	Name	COBOL Stem
AC-CA	Academic College	ACDM-COL
AC-DA	Academic Department	ACDM-DEPT
AC-FA	Academic Division	ACDM-DIV
AC-GA	Academic Term	ACDM-TERM
AC-HA	Academic Term Date	ACDM-TERM-DATE
AC-IA	Academic Term Period	ACDM-TERM-PER
AC-JA	Academic Unit	ACDM-UNIT
AC-KA	Acquisition Date	ACQ-DATE
AC-LA	Acquisition Type	ACQ-TYPE
AC-NA	Actual FTE	ACTL-FTE
AC-OA	Actual Section Size	ACTL-SECT-SIZE
AD-CA	Administrative College or Division	ADMN-COLDIV
AD-DA	Administrative Department	ADMN-DEPT
AD-EA	Administrative Function	ADMN-FNCT
AD-EN	Administrative Unit	ADMN-UNIT
AD-FA	Admission Action Date	ADM-ACT-DATE
AD-GA	Admission Action Reason	ADM-ACT-RSN
AD-HA	Admission Action Type	ADM-ACT-TYPE
AD-HN	Admission Final Action	ADM-FNL-ACT
AD-IA	Admission Level	ADM-LVL
AD-JA	Admission Message	ADM-MSG

Exhibit 3

SOUTHEAST REGIONAL DATA CENTER
FLORIDA INTERNATIONAL UNIVERSITY
IMS DATA INVENTORY
DATE 08/30/73 DATA DICTIONARY/DIRECTORY PAGE AD-EA-01

A. CODE/AD-FA
B. RESPONSIBILITY/
C. LATEST REVISION DATE/09-01-73
D. NAMES
  1. FIU/ADMINISTRATIVE FUNCTION
  2. FAU/
  3. BOR/
  4. WICHE/
E. COBOL NAMES/ADMN-FNCT/XX
F. DEFINITION/THE IDENTIFICATION OF THE INDIVIDUAL FUNCTIONS THAT MAKE UP THE UNIVERSITY FOR BUDGETARY PURPOSES.
G. COMMENTS/COMPONENT OF DEPARTMENT (DE-FA).
H. VALUES
  3. CODES
    A. 00
    B. 01
    C. 02
    D. 03
    E. 04
    F. 05
    G. 06
    H. 07
    I. 61
    J. 62
    K. 63
    L. 71
    M. 72
    N. 73
    O. 74
    P. 75
    Q. 76
    R. 77
    S. 78
    T. 81
    U. 82
    V. 83
    W. 84
    X. 85
    Y. 91
    Z. 92
    AA. 93
  4. DEFINITIONS
    A. ADMINISTRATION
    B. INSTRUCTIONAL
    C. RESEARCH
    D. EXTENSION
    E. LIBRARIES AND MUSEUMS
    F. PLANT OPERATIONS
    G. ORGANIZED ACTIVITIES
    H. NON-EDUCATIONAL

Exhibit 4

SOUTHEAST REGIONAL DATA CENTER
FLORIDA INTERNATIONAL UNIVERSITY
IMS DATA INVENTORY
DATE 08/30/73 DATA DICTIONARY/DIRECTORY PAGE AD-EA-02

- I. WORKING CAPITAL
- J. SELF-SUPPORTING
- K. QUASI-AUXILIARIES
- L. STUDENT ACTIVITIES
- M. SUSPENSE"
- N. FEE ACCOUNTS
- O. STUDENT DEPOSITS
- P. CAMPUS CONCESSION
- Q. ORGANIZATIONAL ACCOUNTS
- R. ATHLETICS
- S. FOUNDATION
- T. SCHOLARSHIPS
- U. FEDERAL LOANS
- V. STATE LOANS
- W. OTHER LOANS
- X. ENDOWMENT
- Y. CONSTRUCTION
- Z. RETIREMENT (INTEREST & SINKING)
- AA. REPAIR AND REPLACEMENT

I. REPORTS WHERE USED
1. AUPO/(DEPARTMENT NO
2. DLDT/(DEPT NO
3. DLSU/(DEPT NO
4. OCSU/(DEPARTMENT NUMBER
5. PDSB/(DEPARTMENT NUMBER

Exhibit 4 Cont'd

SOUTHEAST REGIONAL DATA CENTER
Florida International University
IMS Information Inventory

Report Index

Code	Name	Identifier
ACRC	Acquisitions Receiving Information	IBOLB115
ADRE	Admissions Request	IAOAD211
ADST	Admission Status	IAOAD155
AGDI	Age Distribution	IAOAD210
APOR	Aged Purchase Order	IBOFS252
APST	Application Status	IAOAD202
ARFC	Audit of Registration Fees Collected	
AUPN	Authorized Positions	IBOPR320
AUPO	Authorized Positions	IBOPR320
BUPO	Budget Position	IBOFS195
CNRP	Construction Report	IBOFS184
CCCA	Course Catalog	
CRRF	Name - SS No Cross Reference	IAORG190
DIRE	Distribution of Course Requests	IAOPA001
DLDT	Department Ledger Detail	IBOFS180
DLSU	Departmental Ledger Summary	IBOFS180
DOCO	Daily OCO Purchases	IBOFS160
EDST	External Degree Status	IAOAD152
ENRP	Enrollment Report	
FAFE	Faculty Activity Distribution Summary	IBOPR410
FASC	Faculty Schedule	
FCLS	Fee Card List	
FUAS	Function Assignable	IBOPP208
GNLD	General Ledger	IBOFS200
GNSU	General Ledger Summary	IBOFS205

Exhibit 5

ADP USES OF MINI-COMPUTERS

Alice D. Peters
Digital Equipment Corp.
Maynard, Massachusetts

Lloyd Michalsen
Wheaton College
Wheaton, Illinois

MINICOMPUTERS IN COLLEGE AND UNIVERSITY ADMINISTRATIVE TASKS

My goal today is to very quickly define what a minicomputer is and what its capabilities are currently and for the near future. Then, having established a frame of reference, I will turn the presentation over to Lloyd Michalsen, the Data Processing Director of Wheaton College. Lloyd speaks from his firsthand experience in installing a wealth of administrative applications for the college in Wheaton, Illinois.

What is a Minicomputer?

There are widely divergent opinions as to what constitutes a minicomputer. Current definitions cover products ranging in price from single-board versions of computers costing less than $1,000 to full systems with many peripherals and applications and operating systems software costing more than $100,000. The great range of price tags and diversity of system types can be attributed to several methods of defining a minicomputer. One method is by manufacturer: if a product is made by Company X, it's a minicomputer. A second technique might be by capability. However, since neither of the first two methods withstands all tests, a third and undoubtedly more valid technique is simply by price.

As you probably know, Digital Equipment Corporation originated the idea of the minicomputer in the early 60's. Up to fairly recently, minicomputers were used primarily in dedicated applications such as industrial control and research. However, in the last three to four years the use of mincomputers has been broadened to more general applications and some systems now include a broad range of peripheral devices as well as software for general purpose data processing.

Obviously for administrative processing in higher education we are talking about the latter type of system. Possibly the term small general purpose computer system is more appropriate. By general purpose computer we mean a system complete with necessary mass storage devices (both random and sequential access), card handling devices, and a suitable library of system level and applications software. Until recently the machines satisfying these requirements were the large IBM 360 or 370 computers and smaller general purpose systems such as the IBM 1130 and 1401. However, within the last two years, minicomputer systems have been developed to the point where they possess all the characteristics of the general purpose system. In fact, the computing power and general throughput capability of these minicomputer systems now exceed that of the earlier small general purpose systems and additionally offer exceptional on-line capabilities, yet at a considerably smaller price. Among available systems falling into this category are the PDP-8, PDP-11, H-P 2000,and some D-G systems.

To add further confusion to the computer size classification game, another class of small to medium scale systems, perhaps midicomputer is a good name, is emerging. Systems fitting into this class are PDP-11/45, H-P 3000,and the D-G 840. The advance of hardware technology has permitted this latter class of mini-or midicomputer to offer computational power equivalent to that of traditionally far larger and more expensive systems.

Naturally so many developments in a short time may well cause us to ask what the future holds for minicomputers. As in the past, we expect hardware performance to go up and hardware prices to go down. Contradicting this trend is the increasing cost of software. In the future, users can expect to have a larger array of software offered by minicomputer system suppliers. Additional software offerings will add some incremental cost to the system, but that additional cost will be far less than the in-house cost to develop equivalent software. Future systems software offered by suppliers will also undoubtedly become more terminal-oriented and applications software will no doubt contain better human engineering than has been the case historically.

The next logical question should probably be does the industry trend coincide with where higher education users want to be going. The answer seems to find the two trends in harmony. Users in higher education have been among the most progressive users in adopting the use of terminal-oriented computer systems. Instructional time-sharing--both for computer programming and interactive simulations and gaming--has long been established as a viable tool in the college and university environment. Administrative usage of computers in the batch mode to perform many student and financial accounting jobs including grade reporting, scheduling, transcripts, accounts receivable, accounts payable, and payroll also has a long history in the college and university scene.

However, the wave of the future from both the vendor's, and higher education user's point-of-view, seems to be the system with a dual capability for both administrative and instructional tasks in a timesharing environment. This approach not only offers all the proven advantages of interactive instruction but also provides up-to-the-minute status through on-line inquiry into administrative files Thus, this approach is a cost effective solution to a school's dual requirement for an instructional tool and an administrative system.

To discuss just such a system in detail, I want to introduce Lloyd Michalson, ADP Director at Wheaton College. Although the Wheaton System is used for both administration and instruction, Mr. Michalson's emphasis today will be on the administrative aspects of the Wheaton College Information System.

Mr. Michalsen's paper on the Wheaton College Information System was not received in time for publication. For information on that part of this presentation, please contact Mr. Michalsen.

FEATURE ANALYSIS DEVELOPMENT SYSTEMS

Roger N. Gaunt
Director
Administrative Systems
University of Virginia

David P. Tursan
Manager
Information Systems
University of Virginia

My assignment this morning, at least as I understand it, is to provide an introduction and perspective to the actual Feature Analysis presentation. To facilitate that perspective, I would like to talk for a moment about the attributes of the U.Va. Alumni environment.

Presently, at U.Va. there are twelve (12) decentralized/independent/autonumous entity groups which deal with various aspects of the alumni operation. These groups include the Alumni Associations of the larger professional schools, such as Law, Medicine and Business, as well as the traditional entity groups such as the Alumni Association and the Development Office, plus some that you might not expect, such as "friends of the Library" and the Student Aid Foundation.

The total of active prospects from these twelve alumni entity groups cover 110,000 alumni and friends of the University of Virginia. There are, however, only 52,000 alumni and friends on record for the University. This 112% increase is caused by duplication of information between the alumni entity groups. In particular, each group has created a file of information on alumni who have ties with the group, and on a regular basis these entity groups contact their alumni through mailings, meetings, solicitations, etc. However, since there is (or was) so much redundancy in records management, there are many cases of multiple solicitations, in which the University looks rather awkward to these individuals.

Another completely unrelated attribute of the U. Va. Alumni environment is that presently a significant portion of the U. Va. operating budget comes from alumni and friends. Considering the fact that U. Va. is a state institution this level of support is substantial. Even so, given our new Alumni/Development System, the percentage of operating budget support from alumni and friends is expected to increase. I might add that our expectation of this increase is based on our having talked to several other institutions, particularly Yale, who indicated that once their new system came into operation their gifts per year rose by, as I remember, a million dollars; also, Lehigh pointed out to us that they had uncovered about $50,000, I believe, in pledged by uncollected funds as a result of their file conversion effort (from manual files).

Next, I would like to speak for a moment about the history of U. Va.'s Alumni System Development activity. First, I should mention that some of this activity occured prior to my arrival at U. Va. in August of 1972, so I will have to be brief, or take the risk of being inaccurate. Basically, in approximately 1971, the alumni groups, particularly the Development Office, determined that the groups, particularly the Development Office, determined that the University of Virginia data processing facility did not have sufficient resources to develop the requisite system internally. Accordingly, they went out on the open market to solicit bids for an Alumni/Development System. They captured two: one from Systemation, Inc. and one from Remote Consoles, Inc.

Apparently, at about the time the two bids were received, the Director of the Office of Development died, and for nearly a year thereafter, then lay on the shelf until a new Director of the Office of Development was appointed. Then, several months after the new Director of Development was appointed, the whole computing activities environment at the University of Virginia underwent a major transformation, in which an umbrella organization, called the Office of Computing Activities (OCA) was set up. You will note how this is organized by the organization chart in the handout. The director of this new office felt that it should be possible to develop such a system with internal resources and wanted to use it as the prototype data base system at the University. Accordingly, efforts were begun in this direction. Unfortunately, this individual's enthusiasm for System development did not correspond with the resources he had available or the committments he had already made. This individual's tenure was of rather short duration and ended in his resignation last December for medical reasons.

I point all this out mostly for the purposes of indicating that the Alumni groups had been led along for a period of years by various individuals in the U.Va. hierarchy in the belief that they would have a system "soon".

It was in this context then,that last spring the new director of the Office of Computing Activities, Dr. Donald Wright, and I met with representatives from the Development Office to see what could possibly be done to implement a system by this fall. The results of that meeting were that we agreed that we still wanted to continue to try to use the system as our first Data Base approach and that we would assign Mr. Tursan (who, as a new employee was otherwise uncommitted as well as was appropriate) to the task. Further, we agreed to endorse the idea (of the prior OCA director) to try to import a system from either CAUSE, Yale, or Amherst, and that we would take a "Data Clearinghouse" approach, whereby there would be an Input/Output interface function between the Computer Center and the twelve alumni entity groups. We agreed that we would have a system proposal in the hands of the user by July 2 and (hopefully) a data base system installed and operational in the fall. Finally, we agreed on the table of contents for the system proposal; the Office of Computing Activities agreed to give the system its immediate top developmental priority; and the Development Office agreed to pay for the system with private monies thus substantially speeding up out ability to accomplish this task.

Well, on July 2 the preliminary proposal was delivered. Within 30 days enough agreement existed to begin the file conversion effort, using the DePaul System (a derivative of the Amherst System) as a first phase; incorporating the conceptual design of the Yale reports as a second phase; and finally, using the Amherst Gift Accounting System as a third phase.

By October 15, we had successfully imported the DePaul System (in less than 60 calendar days) with only a half-time graduate student, and 10-man days assistance from Cincom. And, we did it all within 10% of our original cost estimates.

We felt particularly proud of this since it was our first data base system implementation: We knew virtually nothing about the TOTAL package, since it was our first data base installation, and (obviously) nothing about the DePaul System, since we had just acquired it.

Finally, to wrap up my introductory remarks, I would like to briefly review a few characteristics of our Alumni/Development system; it operates under OS on a 370/145 in a 84K partition. In its Phase I configuration, it is basically composed of three software packages. The first is the TOTAL package, which is vended by Cincom Systems in Cincinnati. The second is the GMS package, which we acquired from DePaul University and converted to OS and to ANSI/COBOL, and the third is the RSVP package, which is vended by National Computing Industries, Inc.

From the Phase I system, the following four items are available to the user:

- Edit lists (from GMS)
- Mini-profiles (turnaround maintenance forms), and
- Selective lists and labels (via RSVP)

More details are availabe in the handout.

Now, without any further ado, I will turn the floor over to Mr. Tursan, for the formal Feature Analysis of the five alumni systems contributed to CAUSE.

Thank you very much.

H A N D O U T

CAUSE/December 10, 1973

Introduction to Feature Analysis of Alumni Systems

Roger N. Gaunt

Director, Administrative Systems Division
Office of Computing Activities
University of Virginia

Contents

1. Description of U.Va. Alumni/Development System
2. Center Profile/Fact Sheet of U.Va. Data Processing Function; Organization Structure of Administrative Systems Division

Fact Sheet
Administrative data processing Function
Office of Computing Activities
University of Virginia
Charlottesville, Virginia

Contact:
Roger Gaunt, Director
Administrative Systems Division
4 Dawson's Row

Ker Batton, Assistant Director
Administrative Systems Division
Garrett Hall Annex

Hardware:
IBM 370/145 for Administrative processing only.
Presently 384K bytes of memory, four 3330 disks,
Communications Controller, upper/lower case printer

Software:
OS/MFT/HASP (Operating system with remote, multi-program capability)
TOTAL (Data Base Access Method)
GMS (Generalized File Maintenance System)
WORK-TEN (Cobol Pre-Compiler)
RSVP (English language inquiry System/fast batch)
ANSI COBOL
ISAM (indexed sequential access method)
1460 Emulation

Systems Operational under GMS/TOTAL:

Alumni/Development

Systems Operational under ISAM:

Admissions, Version II (significant enhancement of 72-73 System)
Telephone Billing System

Other Operational 3rd Generation Systems:

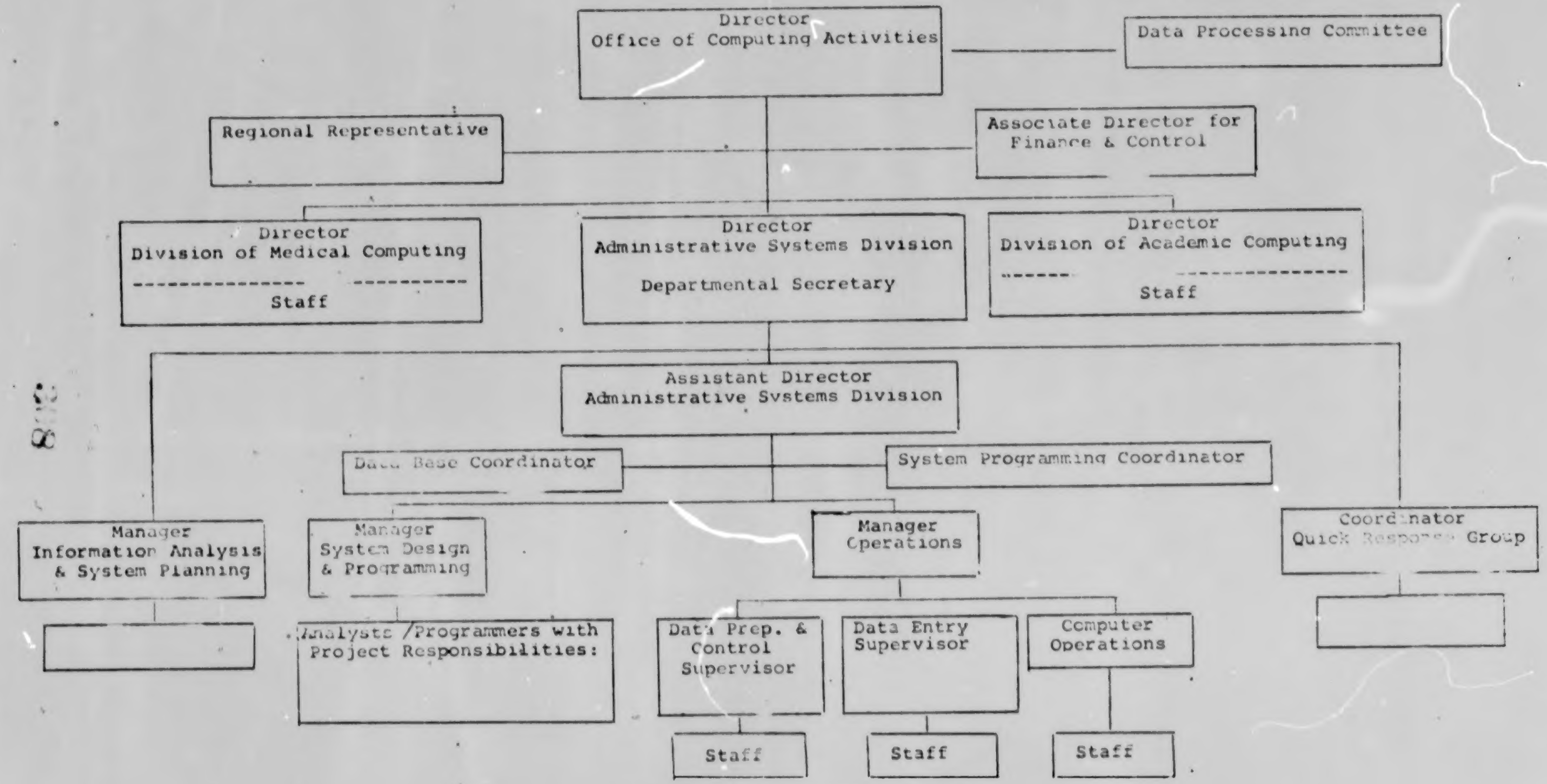
Parking Registration System
Facilities Inventory (F-1)
"Credit-Hour Distribution System" (A-1)
Mailing Label System
WICHE/NCHEMS Simulation models

Systems Being converted to GMS/TOTAL:

Personnel/Payroll File Maintenance
Student Records File Maintenance

Systems Under Development:

Continuing Education Input Processing/Student Registration
Grades Sub-System
Building and Grounds Job Accounting


Director
Office of Computing Activities
Data Processing Committee
Regional Representative
Associate Director for
Finance & Control
Director
Division of Medical Computing
Staff
Director
Administrative Systems Division
Departmental Secretary
Director
Division of Academic Computing
Staff
Assistant Director
Administrative Systems Division
Data Base Coordinator
System Programming Coordinator
Manager
Information Analysis
& System Planning
Manager
System Design
& Programming
Analysts /Programmers with
Project Responsibilities:
Manager
Operations
Coordinator
Quick Response Group
Data Prep. &
Control
Supervisor
Data Entry
Supervisor
Computer
Operations
Staff
Staff
Staff

ADMINISTRATIVE SYSTEMS DIVISION
Organization Chart
November 1, 1973

Alumni/Development System for the University of Virginia

The system uses an IBM 370/145 under OS/MFT/HASP and has been operational, in its phase I configuration since October, 1973. It uses 84K and is composed of twelve (12) programs written in COBOL.

The system is a model of the fourth generation approach to data capture, file maintenance, and information retrieval. In its Phase I configuration, it is a grandson of the Amherst approach and was created by conversion and tailoring of the father system currently operational at DePaul University. One master file is linked in a logical chain with the appropriate detail file(s) in a data base network. The data access method is handled by TOTAL, a proprietary software package of CINCOM Systems, Inc. Information retrieval is handled by RSVP, a proprietary software package of National Computing Industries, Inc.

Major Phase I reports produced are:

- Input Edit Lists (from GMS)
- Mini-Profiles (turn-around maintenance forms)
- Selective Listings (via RSVP)
- Mailing Labels (via RSVP)

Observations on this System

The University of Virginia has recently completed Phase I of a three phased plan to provide its alumni entity groups with a comprehensive system for decentralized alumni operations. During the first phase, information on 53,000 alumni and friends was gathered and recorded in the data base network. Phase I output reports from the data base consist of mailing labels and highly selective listings. It is estimated that 1,200,000 labels will be produced during the first year of operation. Through the efficient operation of the Alumni/Development data base, these mailing labels are estimated to cost less than $0.01 per label. It is also estimated that 180,000 list lines will be generated in the form of "All Alumni who..." reports. Selectivity can be requested at a maximum of 150 condition levels.

During the second phase the primary products to be produced will be periodic reportings as determined by the needs of the twelve alumni entity groups. These groups have different requirements for information from the data base. It was agreed that by providing a highly selective reporting series in Phase I, these alumni entity groups could better plan for the specific periodic reportings needed to manage their individual operations. It is anticipated that many of these reports will follow the design concepts of reports in the Yale Alumni Records Systems (YARDS).

During the third phase the gift reporting sub-system as evidenced in the Amherst (Tri-College) Alumni Records System will be added, via conversion and tailoring, to the Alumni/Development System. It is hoped that activation of the third phase will be scheduled so as to coincide with the review and revision by the University of Virginia of its General Ledger System and its associated chart of accounts.

The approach chosen by the University of Virginia enables each of the 12 alumni entity groups and the Office of Development to administer their operations in a decentralized manner while receiving the cost benefits of a fourth generation data base as well as an organized plan for a tailored, yet general usage alumni records system.

CAUSE/December 10, 1973

Feature Analysis of Alumni Records System

David P. Tursan

Manager, Information Analysis and System Planning

Administrative Systems Division
Office of Computing Activities
University of Virginia

A vast resource of information and assistance exists within the alumni and friends of every institution of higher education. Accordingly, one of the greatest challenges to the Alumni or Development Office function is the identification of the most appropriate and cost effective means of tapping that resource.

Before the resource can be utilized, however, it must first be identified, managed, and monitored in a method paralleling the image and character of the institution. Each and every college or university has a unique set of policies which set the tone for the type of students it attracts, the disciplines in which it specializes, and the means with which it finances its own continuing operations. And with regard to the latter, it is a recognized fact that tuition produces only a portion of the financial resource required to sustain the faculty, staff, facilities, and curriculum of the institution of higher education, and that it is alumni and friends who are increasingly being asked to assist in sustaining them.

On another plane, it is the institution which has as a challenge the identification of the appropriate means of communicating with its alumni and friends. It is this communication through mailings, faculty achievements, and return visits which announces the events that display the policies and image of the university.

To the extent that the challenges of the Alumni or Development function intersect with the challenges of the institution, an Alumni Records System often becomes the central point of university communication to and from its alumni and friends. As a central point, the Alumni Records System needs to serve in multiple roles if it is to be a viable system, and it needs to be able to adjust to new requirements as they are defined.

It is in the identification of these roles that the Alumni or Development function is posed with a series of questions, namely:

-does the university evidence special interest groups such as a Graduate School of Engineering, a Law School, or professional fraternities; do these groups contact alumni and friends?

-is gift-giving administered in a centralized or a decentralized fashion, or is it regionalized via agents who request reportings?

-does the use of "good accounting practices" exist in a a staff of people to process gift information, pledges, and dues?

-is the system for capturing names, addresses, and biographic data adequate to handle the volume of existing alumni, or is it fragmented into multiple files? How are changes of address monitored?

The answers to these questions indicate the direction which will be followed by the Alumni Records system as it assists in meeting the development needs of the particular college/ university.

At this time five Alumni Records systems are available through the CAUSE library. The characteristics represented in those universities operating under these systems are generally as follows:

-The smallest system records data on 30,000 Alumni and Friends while the largest maintains records on 200,000.

-All of the systems carry some version of gift data ranging from a composite figure for gift dollars in this year to a full accounting of the nature, value, and university account receiving the gift.

-Mailing labels, while available in each system, are a primary heavy usage output in one system and are seldom produced in another system.

-The Development Office serves as a primary information source in one system whereas in another eighteen groups can act as an information source.

-One of the systems is currently operating in an "on-line" environment, where data processing (rather than a using department) operates the terminals.

- -As few as fourteen tasks (programs) and as many as fifty-seven tasks (programs) can drive these systems; core storage ranges from 65K bytes to 128K bytes; mass storage varies with one system exclusively using disk, while another primarily uses tape.

- -All systems contain a form of specialized report extraction through specification cards for the production of one-time reports.

From these general characteristics it is evident that these systems present a choice of options for processing data through an Alumni Records System. At the University of Virginia we used the most salient options of a number of these systems as well as options evidenced in the system at Yale University to produce our Alumni/Development System. Based on our experience we feel that the systems in the CAUSE library merit your consideration for your college/university Alumni Records System.

A summary of what we see as the salient features of each of these systems follows.

CAUSE/December 10, 1973

Feature Analysis of Alumni Records System
Part I

SYSTEM CHARACTERISTICS

System	Iowa State	Southern Calif.	Lehigh	Indiana	Amherst
Date of First Production	1969	1970	1970	1968	1971
Computer Utilized	IBM 360/40	IBM 360/40	CDC 6400	IBM 370/155	UNIVAC 70/2
Core Memory Required	128K Bytes	128K Bytes	150K words	N/A	65K Bytes
Primary Storage Media	Tape	Disk	Tape	Disk	Disk
Number of Programs	47 COBOL	12 COBOL 10 Other	30 COBOL 1 Other	29 COBOL 28 Other	14 COBOL
Number of Master Files	3	1	2	1	2
System Generation 1st, 2nd, 3rd, 4th	2nd	3rd	2nd/3rd	2nd/3rd	4th
Documentation Provided (pages)	58	34	150	403	300

CONTINUED ON REVERSE

System	Iowa State	Southern Calif.	Lehigh	Indiana	Amherst
System Emphasis	Member Dues	Data Retrieval	Gift Data	Mailings Information	Gift Accounting
Primary Product	Memorandum of Gift Receipt	Labels	Reports	Labels	Reports
Strength of System	User Procedures	On-line capability	System Design	Decent. Control	Generalized Maintenance
Weakness of System	Multiple Masters	Gift Accounting	Tape Orientation	Gift Accounting	Selective Retrieval
Number of Alumni/Friends	N/A	150,000	38,500	200,000	30,000
Number of Entity Groups	2	1	2	18	3
Number of Campus Sites	1	1	1	11	3
Volume of Data Entries	N/A	N/A	1000 per week	8000 per week	200 , per week
Frequency of Updates	Monthly	Immediate	Weekly	Weekly	Daily
Number of Data Elements	108	61	99	96	75
Selective reports					
user oriented	No	No	Yes	N/A	No
DP oriented	Yes	Yes.	No	somewhat	Yes
Reports visible	N/A	Somewhat	Yes	N/A	Somewhat
Alumni Contacts personalized	No	No	Yes	No	Yes
Class or Regional Agents	No	No	Yes	No	Yes
Gift Data					
-summary/ detail	Both	Both	Both	Sum	Both
-distribution by type of gift	No	No	Yes	No	Yes
-Interface to General Ledger	Yes	No	Yes	No	Yes

CSX-001 Iowa State University

The system uses an IBM 370/40 (128K) under DOS and has been operational since 1969. Its forty-seven (47) programs are written in COBOL. Essentially the system reflects the traditional tape oriented approach with three master files, namely:

Alumni Master Records containing all graduates and friends

Alumni Association Membership Records containing all current members (paid up).

Achievement Fund Records containing year-to-date giving records.

The basic orientation of this system is toward the recording of gift data and membership dues. Two entity groups act as information sources for biographic and gift data. The entity groups are the Alumni Association and the Alumni Achievement Fund.

Updating the master files is scheduled on a monthly basis, and once a year the totals from the Achievement Fund for that year are updated to the Alumni Records Master.

Major reports produced are:

Achievement Fund and Association Cash Reports, listing all cash transactions.

Personnel and/or accounting listings of any group of people desired from Alumni Master.

Pressure sensitive and/or Series 21 Cheshire labels for members of any desired group (from Alumni Master).

Listings in alphabetic, Alumni Number, and Zip Code order (from Alumni Master)

Observations on this system:

One hundred and eight (108) data elements are captured for each alumnus or friend. These elements are common to each of the two individual master files maintained; one by the Alumni Association and one by the Alumni Achievement Fund. Both master files are maintained concurrently during an update run. By using a cross-update operation, Iowa State has provided the multiple entity operations with separate private master file and established means of transferring information collected by one entity group for usage by the other entity group.

In addition, a general record selection via specification cards permits a tape file of desired alumni master records to be used as input to any of the production reports. Hence, although this system lacks a formal report generating program for special purpose reports, the system has the option to produce any of its reports with selective input.

The current Alumni Records System at Iowa State is in Phase II of a three phased plan. Ultimately in Phase III, the multiple masters will be combined as will the paperwork procedures into one simplified bookkeeping and accounting system. The entity groups will remain separate and use common input forms and procedures.

CSX-002 University of Southern California

The system uses an IBM 360/40 (128K) under DOS and has been operational since 1970. It is composed of twelve (12) COBOL and ten (10) site dependent software programs. This system is a third generation on-line approach to file maintenance and information retrieval. One master file, consisting of four segments, is accessed via two (2) model 2260 cathode ray terminals (CRT) on local lines by the Data Processing Mail Files Department.

The basic orientation of this system is toward immediate visual access of a large Alumni Master File. Data describing more than 150,000 alumni and friends is contained on the master file. Any portion of that data can be changed without requiring the traditional approach to data entry, namely: manual transcribing, keypunching, verifying and computer batch updating. As a fail-safe convention, the system can be operated in a batch mode for those few occasions when the on-line capability becomes inoperable.

Major reports produced are:

Four-up mailing tape which is affixed to mailing material by a Cheshire machine.

One-up mailing information.

Donor Statistical Report - Detail Information.

Donor Statistical Report - Summary Information.

3 x 5 File Card.

Computerized letter printed in upper and lower case type.

General Alumni Association Membership Application Card.

One-up pressure sensitive mailing labels used for manually affixing labels of a small volume.

Observations on this system:

Sixty-one (61) data elements are captured for each alumnus or friend. In addition, 432 individual predefined characteristics can be recorded in a matrix of classification codes. This matrix is used to capture facts such as year of graduation, school attended, profession, and affiliation with recognized associations. Through the use of this matrix, the Alumni Association is able to record a significant number of attributes applicable to an alumnus in a simple cost-effective manner. The report selection system is very similar to the one found in the Iowa State University approach in that records can be selected by means of specification cards and reported through any of the production reports.

The most visible attribute of the approach at the University of Southern California is its on-line capability. An interesting sidelight to this on-line capability lies in the location and exclusive usage of the on-line devices by Administrative Data Processing rather than by the end users' department. Normally, one expects the on-line device to be operated by and located in a remote user's office. By taking the approach of locating the devices in data processing and by establishing a procedural line of communication for verbal inquiries or information, the authors of this system provided for security of the information, cost-effective usage of the devices, and a central collection point for all changes.

CSX-003 Lehigh University

The system uses a Control Data Corporation 6400 (150K words) under SCOPE 3.3 and has been operational since 1970. It is composed of thirty (30) COBOL programs and 1 COBOL/COMPASS software program. This system blends together many of the traditional tape oriented conventions, with the third generation design conventions for file maintenance and information reporting. In addition, the system employs exception reporting, good accounting conventions, and full audit trails. Two master files are maintained for each alumnus. One of these, the Alumni System Master file, contains biographic data and summarized gift information for this year and the three previous years. The second master file contains the detail data for each individual gift given to the University by each donor.

The basic orientation of this system is to the accurate and detailed recording of biographic and gift data. In particular, gift data is processed by the system from the point of receipt to the ultimate dispensing of the funds for services within the University. More specifically, gifts of materials, securities equipment, and pledges can be recorded and reported upon by the system.

Updating of both master files is scheduled on a weekly basis.

Forty (40) reports are available, encompassing the following areas:

Progress reports for the current annual giving fund
Prospect reports for area campaigns
Class agent reports
Club area reports
Fraternity reports
Matching gift company reports
Miscellaneous reports

In addition to the above reports, this phase also contains computer mailing capabilities. The selection for the various mailings is on an "All Alumni Who..." basis, using any combination of fields on the Master File. The output of this selection is one of the following:

A non-formated listing
Mailing labels - used in conjunction with a Cheshire-Mod 3

   a. heat seal labels (4 across; 44 labels per page)
   b. heat transfer labels (4 across; 44 labels per page)

Pressure sensitive labels

   a. three across labels (24 labels per page)
   b. four across labels (48 labels per page)

Observations on this system

Ninety-nine (99) data elements are captured for each alumnus or friend. This information is carefully and completely edited upon entry to the system. This system contains a completely documented set of procedures to insure that all information collected is accurate, reported to the responsible entity group, and in the case of gift data, recorded to the General Ledger. The system is set up to make the most complete use of data collected. Lehigh University has a network of class agents and area chairmen responsible for personal contact with the alumnus, as well as gift solicitation. By providing complete and accurate information available to the network of workers, Lehigh University is able to conduct successful campaigns in a professional manner. Of all the systems available through CAUSE, the Lehigh University system seems to this author to be one which evidences the most controlled and documented usage of information. In the case of gifts, the system allows the University to make the maximum effective use of the gift by providing the mechanism for accepting securities, equipment and cash.

The system is well documented and tailored to the individual needs of the departments responsible for maintaining the data on alumni at Lehigh University.

CSX-059 Indiana University

The system uses an IBM 370/155 under OS/MVT and has been operational since 1968. The size of memory required for operation was not provided in the system abstract. Twenty-nine (29) of its programs are written in COBOL; one program is written in ALC, and twenty-seven (27) tasks are sorts or utilities. This alumni records system is designed to capture information on large volumes of alumni for a multi-campus operation. Data applicable to more than 200,000 alumni associated with one of 11 campus sites is recorded on the Alumni Master File tape.

The basic orientation of this system is to the production of mailing labels and accounting for membership dues. Eighteen (18) entity groups act as the information sources for biographic and dues data. These groups represent the Alumni Office, I.U. Foundation, Varsity Club, and 15 sub-entity groups for the academic departments of the University.

Updating of the master file is scheduled on a weekly basis.

Major reports produced are:

- File cards, listing all information maintained for each alumnus
- Association membership cards
- Mailing labels
- Association membership expiration notices
- Geographic and alphabetic listings
- New graduate reports
- Deceased alumni reports

Observations on this system

Ninety-six (96) data elements are captured for each alumnus or friend. In addition to the reports listed above, these elements are used as attributes for the production of mailing labels through a report selection sub-system. The attributes of an alumnus or friend include gift data by year, dues data for affiliated associations, regional campuses attended, and biographic data.

Due to the sheer volume of alumni and friends of Indiana University, the system has been structured to service three important functions.

The first function, mailing labels, is serviced on a daily basis. With 200,000 active alumni, 6,000 new graduates each year and 4,000 deceased alumni each year, the task of communicating with active alumni on a regular basis is highly complex. This system has been structured to produce mailing labels on a selective daily basis in the most cost-effective manner. By these selective mailings, Indiana University is better able to retain a personalized approach with its alumni or friends.

The second function, accounting for membership dues, is handled through weekly postings and monthly expiration notices. Membership may be held in one or more of fifteen sub-entity groups. These groups act as decentralized agents for information transfer. Again, due to the sheer size of the billing and collection operation, Indiana University has chosen to structure its accounting for membership dues in a decentralized manner, through the groups with which the alumni or friend is affiliated.

The third function, file cards for data collected, is handled through the weekly file maintenance run. These file cards are produced as a hard-copy of information, in its current state, recorded for an alumnus or friend. All file cards are retained in central storage by the Alumni Office.

CSX-086 Amherst (Tri-College) College

The system uses a Univac 70/2 (65K) under DOS and has been operational since 1971. Its fourteen (14) programs are written in COBOL. This system is a model for the fourth generation approach to data capture, file maintenance, and information retrieval. One master file is linked in a logical chain with the appropriate detail file in a data base network. The data access method is handled by TOTAL which is a proprietary software package of Cincom Systems, Inc.

The basic orientation of the system is toward capturing and retrieving data. The system has been instructed to address specific elements of data in a category labeled alumni records. Defined characteristics for the specific elements are stored in a disk file as a dictionary. The entity groups who act as information sources represent three separate institutions of higher learning. The entity groups are Amherst College, Mount Holyoke College, and Hampshire College.

Updating of the master file is scheduled to occur as data is available to the system. The system decides if the data applies to its network of alumni information and whether that information is valid for updating the data base.

Major reports produced are:

Gift and Pledge batch lists: Indicate computer edits, is a turn-around document
Gift Distribution Summary: Creates financial accounting summary entries and provides reports as subsidiary ledger.
Agent Reports: Reports class and fund donors and gift activity to agents and workers.
Geographic Reports: Provides geopraphic breakdown of gift and pledge activity for donors and prospective donors.
Range of Giving Reports: Provides dollar range or category breakdown for up to seven accounting selections, e.g. , alumni fund, parents fund, etc.
Gift Matrix Reports: Three one-page reports analyze gifts and pledges by donor type (parent, alumnus, corporation, etc), type of gift (outright, bequests, matching gifts, etc) and by account selection (alumni fund, parents fund, endowment, etc).

Observations on this system

Seventy-five (75) data elements are captured for each alumnus or friend. All maintenance to those data elements is handled through a Generalized Maintenance System. The data elements are stored on a master file of "PEOPLE" established in the data base network. Gifts and pledges are handled through an interface program which balances, validates, and updates the data base network as well as the General Ledger records (not on the Data Base).

As opposed to other approaches to an Alumni Records System, the Amherst System could actually be titled an information handling system for data recognized as applicable to alumni. The data base network contains files comprised of information describing biographic characteristics, pledges, and gift data of alumni. It also contains the linkage to files of information describing the class agents and regional representatives responsible for fund raising campaigns. The gift and pledge data is recorded on files which are linked to the file of alumni.

This approach to alumni records management is controlled by a file labeled "DICTIONARY" and one labeled "TABLE". When data is provided to the system the appropriate data element triggers the system to secure the descriptive characteristics for the element, validate the acceptance conditions of the element, and update the network with the new information.

Through the use of a generalized approach to a data base network, the Amherst system is able to provide the alumni entity groups with important capabilities, namely:

Process gift transactions

Maintain alumni records

Edit batches for valid codes

Produce batch lists showing edit results

Print reports for fund raising campaigns

Provide financial account entries and subsidiary ledgers

Provide lists and mailing labels

Of all the systems available through CAUSE, it is the opinion of this author that the Amherst approach most thoroughly follows "good accounting practices" in its usage of gift data. The system tends to be too accounting oriented rather than people oriented in its means of maintaining alumni information.

DATA INDEPENDENT COMPUTER PROGRAMS AND INFORMATION RETRIEVAL

T. Ray Nanney
Professor of Computer Science
Furman University

Data Independent Computer Programs and Information Retrieval in Small Educational Institutions

T. Ray Nanney, Furman University
December 1973

The use of data independent computer programs has been found to be particularly helpful in the understaffed computer centers typically found in small universities and small colleges. The validity of this point will be demonstrated by summarizing the experiences of several institutions with one data independent program, the National Laboratory for Higher Education Information Retrieval System (NLHE IRS). Then some comments will be made regarding the future use of data independent programs.

The plight of the small computer center of an educational institution has been discussed so many times in the literature as to be almost a cliché. Nevertheless, some important difficulties in such centers will be summarized here in preparation for the discussion of data independent programs. Many of the computer centers, of which I have personal knowledge, have no more than three programmers--most have a computer center director and a programmer. These individuals are under constant pressure to do a significant amount of work for administrative offices in order to justify the high cost of equipment. When a computer center is first established, there is ordinarily considerable success in implementing computer based applications for administrative offices. Then, in the dynamic environment of small institutions--it is relatively easy to change academic programs and administrative procedures in small institutions--more and more time must be devoted to changing the existing computer programs to make them conform to new situations. Progress on new computer programs slows and may eventually stop. Because of the success of the initially installed computer programs, administrative offices apply pressure to have new systems implemented. The computer center director is in a dilemma: he has sufficient personnel to modify existing programs or to implement new programs, but both cannot be done. Yet, the future success of the computer center depends on being able to do both.

Data independent computer programs represent one method of escaping from the situation just described. Normally, a data independent program will be designed with sufficient flexibility to accommodate significant changes in the data files it manipulates. This characteristic reduces the program maintenance problem to a manageable level, and the computer center personnel escape from the dilemma. Also, information retrieval systems and data management systems can be used to reduce the effort of developing new computer applications.

Unfortunately, data independent computer programs are more difficult to write than ordinary computer programs, a longer time is required to write them, and their initial cost is greater. For example, a much greater effort is required to write a payroll program which will work satisfactorily for most small educational institutions than to write one which works for a particular institution. Note, however, that the school which has a generalized payroll program will probably be able to change its payroll procedures several times without changing the computer program in any significant way. The development cost of one good data independent program is likely to be less than the cost of a corresponding ordinary program plus its subsequent modifications.

A computer program written in data independent form will contain extra instructions relative to a program tailored for a particular situation. This is surely a disadvantage of generalized programs. Experience has shown, however, that the carefully written generalized program may execute in the same or even less time than specialized programs written rapidly because of pressures on the computer center.

The NLHE Information Retrieval System is an example of a data independent computer program which is used by many educational institutions. It was written by personnel at Clarke College, Dubuque, Iowa, and at Furman University, Greenville, South Carolina, in response to the needs of a group of IBM 1130 computer users who had been brought together by NLHE in January 1970 to discuss their common problems. This group became inspired by the work of William Verbrugge and his associates at Saint Joseph's College in Rensselaer, Indiana, who were the first, to our knowledge, to successfully implement an information retrieval system on a computer as small as the IBM 1130. In 1970 the commercially available information retrieval systems, for example, the MARK series of Informatics, Inc., were usable only on large computers and also were too expensive for small educational institutions.

From the beginning the development and use of the NLHE Information Retrieval System has been a learning experience for those individuals directly involved in the project. After an initial development period, a selected group of users furnished feedback with respect to additional features they wanted incorporated into the programs. When the system was made generally available in 1971, it contained all the additional features which had been requested by the initial users. In February 1972 NLHE conducted a survey as part of the evaluation of the system. The survey provided additional insight regarding modifications to the system which would make it even better for small educational institutions. Results of the evaluation will be given later.

The NLHE Information Retrieval System, our example of a set of data independent programs, was designed to

1. provide an integrated set of programs for generating, updating, and maintaining disk data files.
2. allow sorting of the disk file.
3. select subsets of information stored in a disk file.
4. print any desired fields in the order required for an arbitrary report after the sorting or selecting of records from the disk file.
5. generate statistical reports on the total file or subsets of the file.
6. be used with ease by persons unfamiliar with computer programming, so that reports can be generated without requiring the services of a programmer.

The system also included an easy-to-use report generating feature for reports requiring both strings of literal information and data from the disk file. The programs are driven by a table, the DESCRIPTION TABLE, which is stored as a disk data file. Within reasonable limits, the NLHE Information System will retrieve information from any disk data file provided that a DESCRIPTION TABLE has been stored previously.

As examples, DESCRIPTION TABLES are given in Figures 1 and 2 for Furman University's Development Office Current Gift File and its Pledge File. Each line of these tables is a description of a data field which can be manipulated by the NLHE IRS. For a given data file, if new data fields are introduced or existing fields are to be redefined, it is only necessary to create a new DESCRIPTION TABLE to be able to use the NLHE IRS on the modified file.

Figure 1

DEVELOPMENT OFFICE CURRENT GIFT FILE •DVGFT•

FIELD DESCRIPTION	ID CODE FOR SORTING AND/OR SELECTING	ID CODE FOR PRINTING	NUMBER OF PRINT POSITIONS	COLUMN HEADINGS ON REPORT	EDIT WORDS FOR PRINT FIELDS	FACTORS USED IN STATISTICAL ROUTINES
DONOR IDENTIFICATION NUMB		1	7	ID NUMB		
I.D.NO.-SEX OR TYPE	1					
I.D.NO.-CLASS	2					
I.D.NO.-ALPHA	3					
FIRST NAME	4	2	12	FIRST NAME		
INITIAL	5	3	1	I		
LAST NAME	6	4	15	LAST NAME		
LEDGER ACCOUNT NUMBER	7	5	7	ACCT	' - '	
AMOUNT OF GIFT	8	6	12	AMOUNT	' , , 0. '	0.01
DATE OF TRANSACTION		7	8	DATE	' / / '	
DATE- MONTH	9					
DATE- DAY	10					
DATE- YEAR	11					
RECEIPT NUMBER	12	8	5	RECPT		
PROGRAM CODE	13	9	2	PG		
TYPE OF GIFT	14	10	1	T		
JOINT GIFT	15	11	1	J		
MATCHABLE GIFT	16	12	1	M		
JULIAN DATE	17	13	3	JND		
PAYMENT ON PLEDGE	18	14	1	P		

DEVELOPMENT OFFICE PLEDGE FILE •DVPLG•   Figure 2

FIELD DESCRIPTION	ID CODE FOR SORTING AND/OR SELECTING	ID CODE FOR PRINTING	NUMBER OF PRINT POSITIONS	COLUMN HEADINGS ON REPORT	EDIT WORDS FOR PRINT FIELDS	FACTORS USED IN STATISTICAL ROUTINES
DONOR IDENTIFICATION NUMB		1	7	ID NUMB		
I.D.NO.-SEX OR TYPE	1					
I.D.NO.-CLASS	2					
I.D.NO.-ALPHA	3					
FIRST NAME	4	2	12	FIRST NAME		
INITIAL	5	3	1	I		
LAST NAME	6	4	15	LAST NAME		
LEDGER ACCOUNT NUMBER	7	5	7	ACCT	' . '	
AMOUNT OF GIFT	8	6	12	AMOUNT	' , , 0 . '	0.01
DATE OF PLEDGE		7	5	DATE	' / '	
DATE-MONTH	9					
DATE-YEAR	10					
UNUSED	11	8	2			
PROGRAM CODE	12	9	2	PG		-
QUARTERS REMAINING TO PAY	13	10	2	QR		
AMOUNT DUE THIS QUARTER	14	11	10	1ST QTR	' , , 0 .'	1.00
AMOUNT DUE 2ND QUARTER	15	12	10	2ND QTR	' , , 0 .'	1.00
AMOUNT DUE 3RD QUARTER	16	13	10	3RD QTR	' , , 0 .'	1.00
AMOUNT DUE 4TH QUARTER	17	14	10	4TH QTR	' , , 0 .'	1.00
AMOUNT DUE 5TH QUARTER	18	15	10	5TH QTR	' , , 0 .'	1.00
AMOUNT DUE 6TH QUARTER	19	16	10	6TH QTR	' , , 0 .'	1.00
AMOUNT DUE 7TH QUARTER	20	17	10	7TH QTR	' , , 0 .'	1.00
AMOUNT DUE 8TH QUARTER	21	18	10	8TH QTR	' , , 0 .'	1.00
AMOUNT DUE 9TH QUARTER	22	19	10	9TH QTR	' , , 0 .'	1.00
AMOUNT DUE 10TH QUARTER	23	20	10	10TH QTR	' , , 0 .'	1.00
AMOUNT DUE 11TH QUARTER	24	21	10	11TH QTR	' , , 0 .'	1.00
AMOUNT DUE 12TH QUARTER	25	22	10	12TH QTR	' , , 0 .'	1.00
AMOUNT DUE 13TH QUARTER	26	23	10	13TH QTR	' , , 0 .'	1.00
AMOUNT DUE 14TH QUARTER	27	24	10	14TH QTR	' , , 0 .'	1.00
AMOUNT DUE 15TH QUARTER	28	25	10	15TH QTR	' , , 0 .'	1.00
AMOUNT DUE 16TH QUARTER	29	26	10	16TH QTR	' , , 0 .'	1.00
AMOUNT DUE 17TH QUARTER	30	27	10	17TH QTR	' , , 0 .'	1.00
AMOUNT DUE 18TH QUARTER	31	28	10	18TH QTR	' , , 0 .'	1.00
AMOUNT DUE 19TH QUARTER	32	29	10	19TH QTR	' , , 0 .'	1.00
AMOUNT DUE 20TH QUARTER	33	30	10	20TH QTR	' , , 0 .'	1.00
LAST AMOUNT PAID ON PLEDG	34	31	12	LAST PD	' , , 0 . '	0.01
DATE OF LAST PAYMENT		32	5	DLPM	' /0 '	
MONTH OF LAST PAYMENT	35					
YEAR OF LAST PAYMENT	36					
BALANCE DUE ON PLEDGE	37	33	12	BAL DUE	' , , 0 . '	0.01
AMOUNT PAST DUE	38	34	12	PAST DUE	' , , 0 . '	0.01
AMOUNT PAID THIS QUARTER	39	35	12	PD IN QTR	' , , 0 . . '	0.01

The columns headed "ID Code for Sorting And/Or Selecting" and "ID Code for Printing" contain codes used in the instructions for the NLHE Information Retrieval System. For example, for the current gift file, the command

```
PRINT 4 3 6 7
```

requests that a donor's last name, initials, gift, and date of the gift be printed in this sequence on a line. For the same file, the commands

```
SORT 6 ASCENDING
SORT 5 ASCENDING
```

request that records be sorted alphabetically by last name and then by initials. Figure 3 is a summary of the information retrieval commands of the system. To give some insight into the power of the system, several sample reports are given in the appendix.

In mid-February 1972, approximately nine months after beginning general distribution of the system, NLHE conducted a survey to determine the impact of the system on educational users. Questionnaires were mailed to 190 educational institutions who had been sent the system. A total of 75 questionnaires were returned including 20 from institutions which had not installed the system. A summary is given in Figure 4 of the characteristics of the 55 institutions which had installed the system. The majority were private institutions having fewer than 2000 students. Thirty institutions reported installing the system in 3 days or less (Figure 5). A majority of the institutions, a total of 33, reported having two or more working applications of the system. See Figure 6 for a summary of the number of applications reported. Comments made about the system were overwhelmingly favorable.

Figure 3

FREE FORMAT INFORMATION RETRIEVAL COMMANDS OF THE NLHE INFORMATION SYSTEM

COMMAND	ACTION TAKEN
DATE	Print a date and the optional report identification information at the top of each page of the report. Example: DATE 120774 ADMISSIONS OFFICE
SELECTION	Selecta subset of records. For statistical computations a subset and subsets of it can be manipulated in a single run. Example: SELECTION CRITERIA = 2 FIELD = (A,10), LIMITS = ('B','B') FIELD = (B,50), LIMITS = ('2','2') A AND B ACCEPTED BOARDING STUDENTS
SORT	Sort ascending or descending on a field. A maximum of 10 SORT cards can be used in a run. Example: SORT 4 ASCENDING
PRINT	Print a line of data fields from each record of interest. Example: PRINT 1 2 3 6 40
FORMAT	Provide line spacing and control break information for the PRINT command. Example: FORMAT DOUBLE SPACE CONTROL BREAK
REPRINT	Use the last sort and/or selection of records for printing this report. Example: REPRINT
TABULATE	Count occurrences of a given value for those fields which are sorted during this run. Example: TABULATE

Figure 3 - (cont.)

COMMAND	ACTION TAKEN
CORRELATION	Calculate and print correlation coefficients and supplementary statistics. A maximum of ten correlations can be done in a given run. Example: CORRELATION 42 45
FREQUENCY	Calculate and print a frequency distribution table and supplementary statistics. A maximum of five frequency distributions can be specified for each selected subset in a run. Example: FREQUENCY 42 200. 50. 13
EXIT	Exit immediately after sorting and selecting to allow processing of these records by an auxiliary program. Example: EXIT
END	Signal the end of the information request disk to the controlling program. Example: END

Figure 4

CHARACTERISTICS OF INSTITUTIONS RETURNING
NLHE INFORMATION SYSTEM SURVEY FORM

(Institutions Having System Installed)

Type Institutions	Number of Institutions
High school or public school system	6
Junior or community college	5
Private college or university	37
Public college or university	7
Total	55

Number of Students Enrolled	Number of Institutions
less than 1000	12
1001 - 1500	14
1501 - 2000	17
2001 - 2500	5
greater than 2500	7
Total	55

Figure 5

RESPONSES TO THE QUESTION

"How long did it take you to install the NLHE system?"

NLHE INFORMATION SYSTEM SURVEY FORM
(Institutions Having System Installed)

Time	Number of Institutions
less than 1 day	6
1 day	9
2 days	9
3 days	6
4 days	1
more than 1 week	20
no answer	4
Total	55

The minimum installation time reported was 2 hours, and the maximum was 4 months.

Figure 6

NUMBER OF APPLICATIONS REPORTED

NLHE INFORMATION SYSTEM SURVEY FORM
(Institutions Having System Installed)

Number of Applications	Number of Institutions
0	3
1	19
2	11
3	9
4	3
5	4
6	2
7	2
8	0
9	1
10	1
Total	55

Twenty-seven distinct applications of the NLHE Information System have been reported by educational institutions. A list of these applications is given in Figure 7. The variety of these applications plus the fact that the majority of reporting institutions had two or more operating applications, jointly constitute strong evidence for the validity of the earlier claim: data independent computer programs are particularly helpful in small, understaffed computer centers. Respondents to the survey stated the NLHE Information System

1. has powerful facilities for generating reports.
2. assists in handling unanticipated requests for reports.
3. reduces the number of "one-time" programs which must be written.
4. increases the efficiency of computer center operations.
5. increases the speed with which data processing tasks can be handled.
6. is very easy to use.

Several suggestions for improvements to the system were made by respondents. Three individuals indicated that the sorting speed could be improved, and methods were developed subsequently to accomplish this. The ability to do arithmetic computations was mentioned several times as a highly desirable feature. Automatic table-lookup of encoded information was also requested. Users of slow printing equipment requested a feature to print multiple side-by-side forms, that is, 2- and 3-up forms.

During the period October 1972 through March 1973, NLHE conducted an indepth study of the use of its information system at five institutions. All use of the NLHE IRS was noted, and copies of reports were retained for mailing to NLHE. On the average, each institution produced 290 reports during the six months of study (1.3+ reports per working day per institution).

Figure 7

REPORTED APPLICATIONS OF NLHE INFORMATION RETRIEVAL SYSTEM IN EDUCATIONAL INSTITUTIONS

1. Admissions information system
2. Alumni information system
3. Applications for financial aid
4. Asset control system
5. Athletics office mailings
6. Building fund data
7. Church/Counselor data
8. Course reservations
9. Development office data
10. Enrollment statistics
11. Faculty data file
12. Football scouting reports
13. Grade reporting
14. Institutional research
15. Inventory control and reporting
16. Library acquisitions
17. Music and Art for the Disadvantaged
18. Payroll reporting
19. Personnel guidance
20. Preregistration
21. Registration
22. Research in Folk Religion
23. Serials in library
24. Space utilization
25. Student information system
26. student profile
27. Union Catalog of Library Holdings

The number of reports produced by individual institutions is given in Figure 8. These data further emphasize the value of data independent computer programs in the small computer center.

Data independent information retrieval systems and data management programs are now fairly common for large computers. In December 1972, more than 75 different information systems were being marketed with their purchase prices ranging from approximately $2800 to $120,000. Data independent programs are not limited to information retrieval; for example, such generalized programs can be purchased for almost any accounting function of interest to business organizations.

The development of additional, even more powerful data independent computer programs for small educational institutions appears to be both desirable and inevitable. Methods are now known to allow writing such a program for almost any administrative activity within a college or university. Furthermore, these programs can be written in a manner to allow their execution on small computers. For example, NLHE is now distributing computer programs for a generalized General Ledger and Budgetary Accounting System which appears to be suitable for either large or small educational institutions and which can be run on the small IBM 1130 computer. It is anticipated, therefore, that generalized programs will be used by both small and large educational institutions.

Because of the cost of writing data independent programs, most of the large, complex programs probably will be produced by software development companies. Such a procedure will allow the cost of a program to be distributed among its purchasers. It is hoped that those institutions which do develop their own data independent programs will share them with other institutions through organizations like CAUSE.

Figure 8

NUMBER OF REPORTS GENERATED USING NLHE INFORMATION SYSTEM

October 1972 - March 1973

Institution	Enrollment	Number of Applications	Number of Reports
Concord College Athens, W. V.	1843	11	687
Furman University Greenville, S. C.	2089	9	133
George Peabody College Nashville, Tenn.	1980	3	26
Muskingum College New Concord, Ohio	1168	5	299
Oklahoma Baptist University Shawnee, Okla.	1701	6	309

FURMAN ADMISSIONS FILE 70-71   FORMAT AS OF FEBRUARY 12, 1971

FIELD DESCRIPTION	ID CODE FOR SORTING AND/OR SELECTING	ID CODE FOR PRINTING	NUMBER OF PRINT POSITIONS	COLUMN HEADINGS ON REPORT
SOCIAL SECURITY NUMBER	1	1	11	SOC SEC
1ST & 2ND NAME OR INITIA	2	2	13	NAME
LAST NAME	3	3	15	LAST NAME
INITIALS	4	4	2	IN
PREFERRED NAME	5	5	9	PREF NAME
SEX	6	6	1	S
APPLICANT BIRTHDAY	7	7	8	BIRTH
PLACE OF BIRTH	8	8	15	BRTH PLACE
PLANNED DATE OF ENROLLMT	9	9	3	DE
BOARDING OR COMMUTING	10	10	1	B
TYPE-NEW,TRANS,RE-AC	11	11	1	T
EARLY DECISION	12	12	1	E
CAREER CHOICE	13	13	2	CR
APPLICANT DENOMINATION	14	14	4	DENM
APPLICANT HOME ADDRESS	15	15	23	ADDRESS
CITY	16	16	13	CITY
STATE	17	17	2	ST
ZIP CODE	18	18	5	ZIP
SC COUNTY CODE	19	19	2	CC
HOME PHONE	20	20	12	PHONE
MARITAL STATUS	21	21	1	M
NAME OF SPOUSE	22	22	9	SPOUSE
NUMBER OF CHILDREN	23	23	1	C
FOREIGN STUDENT	24	24	1	F
ALUMNI RELATED	25	25	1	A
PARENT OR GUARDIAN CODE	26	26	1	P
PARENT OR GUARDIAN NAME	27	27	23	P/G NAME
P/G OCCUPATION	28	28	10	OCCUPATION
P/G EMPLOYER	29	29	30	EMPLOYER
HIGH SCHOOL F/W GRADUATD	30	30	15	HIGH SCHL
CEEB HS CODE	31	31	6	CEEB
TYPE HIGH SCHOOL	32	32	1	H
FURMAN SCHOLAR	33	33	1	U
NAT'L MERIT SCHOLAR	34	34	1	N
LEADERSHIP	35	35	2	LD
LITERATURE-EDITOR	36	36	1	L
MUSIC PARTICIPATION	37	37	1	M
ATHLETICS	38	38	2	AT
LAST COLLEGE ATTENDED	39	39	15	LAST COLL
CLASS RANK	40	40	4	RANK
CLASS SIZE	41	41	4	SIZE
SAT-VERBAL	42	42	4	STV
SAT-MATH	43	43	4	STM
WEEK APPLICATN RECEIVED	44	44	3	WK
PREDICTIVE GPA	45	45	4	GPA
LATEST CLASS RANK	46	46	4	LRNK

CONTINUED ON NEXT PAGE

CONTINUED
FURMAN ADMISSIONS FILE 70-71   FORMAT AS OF FEBRUARY 12, 1971

FIELD DESCRIPTION	ID CODE FOR SORTING AND/OR SELECTING	ID CODE FOR PRINTING	NUMBER OF PRINT POSITIONS	COLUMN HEADINGS ON REPORT
LATEST CLASS SIZE	47	47	4	LSIZ
HIGHEST SATV	48	48	4	HV
HIGHEST SATM	49	49	4	HM
BOARD DECISION	50	50	1	D
APPLICANT RESPONSE	51	51	1	R
FINANCIAL AID	52	52	1	F
TYPE AID	53	53	2	TA
CONVERTED RANK IN CLASS	54	54	4	CON
LOADED ON SIS DISK	55	55	1	L
ENROLLED FALL TERM	56	56	1	E
HIGHEST SAT TOTAL SCORE	57	57	5	HTOT

```
          LISTING OF INPUT CONTROL CARDS
DATE     011572       SAMPLE REPORT 13
SORT   3
SORT   4
PRINT   1 2 3 6 40 41 42 43 45  50 54  57
SELECTION     CRITERIA = 1
FIELD =(A,50),  LIMITS=('2','2')
A
APPROVED APPLICANTS
END



                    SELECTIVITY INSTRUCTIONS

APPROVED APPLICANTS

BOOLEAN EXPRESSION            A
     BOUNDS SET 'A' IS    BOARD DECISION
          LOWER LIMIT      2
          UPPER LIMIT      2
```

Example 1: Listing of approved applicants

(Page 1 of 4)

```
THIS REPORT WAS GENERATED ON 01/15/72.
SAMPLE REPORT 13

THE FOLLOWING ARE THE COLUMN HEADINGS AND AN EXPANDED EXPLANATION

COLUMN  FIELD NO  HEADING      EXPLANATION

   1        1     SOC SEC      SOCIAL SECURITY NUMBER
   2        2     NAME         1ST & 2ND NAME OR INITIA
   3        3     LAST NAME    LAST NAME
   4        6     S            SEX
   5       40     RANK         CLASS RANK
   6       41     SIZE         CLASS SIZE
   7       42     STV          SAT-VERBAL
   8       43     STM          SAT-MATH
   9       45     GPA          PREDICTIVE GPA
  10       50     D            BOARD DECISION
  11       54     CON          CONVERTED RANK IN CLASS
  12       57     HTOT         HIGHEST SAT TOTAL SCORE


 THE NUMBER OF RECORDS PRINTED ON THIS REPORT IS    101
```

Example 1: Listing of approved applicants

(Page 2 of 4)

```
DATE 01/15/72                                                  PAGE   1
APPROVED APPLICANTS

SOC SEC     NAME             LAST NAME        S RANK SIZE STV  STM  GPA  D CON  HTOT

000-71-0073 THOMAS E         AAAAK            M   11  139 531. 437. 2.01 2  7.5  968.
000-71-0074 JOSEPH D         AAAAL            M                .    .    .  2   .    0.
000-71-0075 GERHARDT M       AAAAM            M    8   36 490. 540. 1.96 2 20.8 1030.
000-71-0078 LAWRENCE D       BBBBB            M   33  264 550. 630. 2.32 2 12.3 1180.
000-71-0079 EDSEL WAYNE      BBBBC            M   10   45 700. 470. 2.09 2 21.1 1170.
000-71-0080 JACQUELINE       BBBBD            F   32  300 700. 729. 2.78 2 10.5 1429.
000-71-0085 SAU CHUN         BBBBH            F                .    .    .  2   .    0.
000-71-0092 ROBERT B         BBBBN            M   20   65 650. 670. 2.33 2 30.0 1320.
[illegible]-1567 JANE AVERY  BBBBR            F * 36  231 438. 352. 1.68 2 15.3  790.
[illegible]-5697 DAVID TYLER BBBBX            M    1   45 480. 590. 2.57 2  1.1 1070.
[illegible] MARK DAVID       BBBBZ            M   43  162 423. 726. 2.15 2 26.2 1149.
[illegible]-5527 JOHN THOMAS CCCCA            M    7  747 505. 570. 2.56 2   .8 1075.
[illegible]-6961 [illegible] M CCCCC          M   21   33 493. 661. 1.79 2 62.1 1154.
[illegible]-3051 SHERRI LYNN CCCCN            F   74  342 670. 680. 2.51 2 21.4 1350.
[illegible]-1460 DAVID DENIS CCCCO            M    8   60 510. 520. 2.05 2 12.5 1030.
[illegible]-6156 CLINTON T   CCCCP            M  126  444 540. 633. 2.11 2 28.2 1173.
[illegible]-9450 SHARON LEE  CCCCQ            F   11  390 610. 600. 2.77 2  2.6 1210.
[illegible]-0423 DAVID MICHAEL CCCCT          M   35  342 487. 581. 2.14 2 10.0 1068.
[illegible]-0647 DAVID CHARLES CCCCU          M   55  204 470. 510. 1.77 2 26.7  980.
[illegible]-7856 ELIAS       DDDDA            M  209  550 628. 600. 2.05 2 37.9 1228.
[illegible]-7122 DANIEL A    DDDDC            M    7   70 480. 536. 2.04 2  9.2 1016.
[illegible]-9525 BRIAN MICHAEL DDDDE          M    6  245 440. 570. 2.37 2  2.2 1010.
[illegible]-5501 KAREN M     DDDDF            F   63  310 450. 556. 1.99 2 20.1 1006.
[illegible]-8223 VERONICA M  DDDDH            F    8   63 310. 366. 1.66 2 11.9  676.
[illegible]-7830 JAMES THOMAS DDDDI           M   55  127 674. 595. 2.10 2 42.9 1269.
[illegible] ALAN CLAY        DDDDJ            M   94  615 450. 640. 2.11 2 15.2 1090.
[illegible]-9336 DAVID PHILIP DDDDM           M   10   79 493. 680. 2.35 2 12.0 1173.
[illegible]-44-7410 VERA E   DDDDO            F   55  174 500. 400. 1.69 2 31.3  900.
[illegible]-46-2031 JANE IDA DDDDQ            F    1   27 580. 620. 2.91 2  1.8 1200.
[illegible]-42-3219 JANET    DDDDR            F   68  342 569. 650. 2.31 2 19.7 1219.
[illegible]-46-2445 HOWARD T DDDDS            M    8   55 380. 300. 1.43 2 13.6  680.
[illegible]-45-8474 GARRY W  DDDDU            M  234  386 394. 380. 1.09 2 60.4  774.
[illegible]-38-3239 LARRY LEROY DDDDV         M   79  391 516. 550. 2.01 2 20.0 1066.
117-48-1235 PAUL J           EEEEA            M  268  610 628. 510. 1.87 2 43.8 1138.
121-58-6362 MICHAEL HAYS     EEEEH            M  134  245 384. 500. 1.43 2 54.4  884.
121-46-7863 MARY HAZEL       EEEEI            F   18  175 660. 650. 2.60 2 10.0 1310.
122-46-5179 MARK KEITH       EEEEJ            M   80  309 486. 613. 2.00 2 25.7 1099.
123-46-6858 MARILYN B        EEEEK            F    4  607 710. 680. 3.50 2   .5 1390.
125-44-1059 BALDWIN          EEEEL            M   38   98 530. 442. 1.49 2 42.6  972.
12[illegible]-46-9506 EDWARD T EEEEM          M   57  446 510. 470. 1.95 2 12.6  980.
13[illegible]-40-8202 EDMUND A EEEER          M    7  333 595. 670. 2.77 2  1.9 1265.
138-43-5576 ALICE V          EEEES            F   37  127 690. 566. 2.22 2 28.7 1256.
141-50-6512 JAY B            EEEEW            M   81  256 550. 560. 1.97 2 31.4 1110.
143-38-2827 MICHAEL A        FFFFB            M  104  131 628. 729. 1.90 2 79.0 1357.
143-48-3524 CARL JOSEPH      FFFFC            M   75  366 618. 700. 2.45 2 20.3 1318.
144-56-0137 ROBERT C         FFFFD            M   35  260 461. 630. 2.21 2 13.2 1091.
14[illegible]-40-7338 STEVEN E FFFFF          M   36  820 512. 545. 2.31 2  4.3 1057.
147-38-3138 TERESA LOUISE    FFFFI            F   76  559 720. 700. 2.77 2 13.5 1420.
150-44-0856 ERIC STUART      FFFFM            M   34  212 610. 730. 2.50 2 15.8 1340.
150-48-4372 EDWARD J         FFFFN            M  102  402 404. 550. 1.76 2 25.2  954.
153-48-3992 MICHAEL F        FFFFS            M   43  236 483. 530. 1.93 2 18.0 1013.
154-44-7429 JAMES            FFFFY            M   88  336 407. 570. 1.81 2 26.0  977.
155-46-7312 NANCY FORREST    FFFFZ            F    3  287 680  610. 3.20 2   .8 1290.
156-48-1472 KEVIN T          GGGGA            M   60  654 500. 730. 2.46 2  9.0 1230.
157-36-7651 THOMAS WARREN    GGGGB            M  177  430 376. 518. 1.56 2 41.0  894.
```

Example 1: Listing of approved applicants

(Page 3 of 4)

```
DATE 01/15/72                                                  PAGE   2
APPROVED APPLICANTS

SOC SEC     NAME             LAST NAME        S RANK SIZE STV  STM  GPA  D CON  HTOT

158-38-9340 WILLIAM T        GGGGC            M   80  143 608. 770. 2.27 2 55.5 1378.
157-46-2491 JERRY ALAN       GGGGD            M  120  402 410. 540. 1.75 2 29.7  950.
163-46-2969 ELIZABETH ANN    GGGGG            F    9  264 641. 530. 2.72 2  3.2 1171.
164-46-6045 DONALD A         GGGGH            M  302  505 487. 660. 1.78 2 59.7 1147.
167-44-4806 CLINTON RAY      GGGGJ            M    6  191 681. 777. 3.00 2  2.8 1458.
168-42-4960 WOODROW W        GGGGK            M   58  220 400. 670. 2.00 2 26.1 1070.
170-46-6099 WILLIAM E        GGGGO            M  198  506 659. 563. 2.02 2 39.0 1222.
171-38-1225 RICHARD SCOTT    GGGGP            M   74  174 540. 538. 1.81 2 42.7 1078.
176-44-6228 ROBERT ALTON     GGGGS            M   23  230 480. 633. 2.24 2  9.7 1113.
177-46-5046 WINFRED C        GGGGT            M  107  344 580. 630. 2.16 2 30.9 1210.
177-42-2318 JEFFREY JAY      GGGGU            M  224  520 580. 530. 1.85 2 42.9 1110.
179-42-4812 ANDREW W         GGGGV            M   15   54 550. 730. 2.32 2 26.8 1280.
181-38-6954 GERALD P         GGGGX            M   55  100 512. 590. 1.78 2 54.5 1102.
182-46-0590 JANET MARIO      GGGGY            F    2  852 733. 738. 3.78 2   .1 1471.
190-46-2314 DANIEL G         HHHHE            M   25  153 600. 620. 2.26 2 16.0 1220.
190-46-2810 WILLIAM C        HHHHF            M   65  149 510. 599. 1.90 2 43.9 1109.
205-40-4132 ELEANOR M        HHHHM            F    5  303 681. 566. 2.98 2  1.6 1247.
208-42-6751 JANICE LEE       HHHHP            F  205  521 407. 499. 1.53 2 39.2  906.
210-36-0393 BLAINE JOSEPH    HHHHQ            M   65  308 506. 590. 2.08 2 20.9 1096.
211-44-0794 KATHLEEN M       HHHHS            F    4  264 560. 624. 2.57 2  5.1 1184.
211-44-1493 DONALD ALAN      HHHHT            M    7  529 667. 767. 3.06 2  1.2 1434.
212-58-3262 JAMES J          HHHHV            M    4   54 371. 560. 2.05 2  6.4  931.
212-58-6252 THADDEUS W       HHHHW            M   45   96 621. 600. 1.94 2 46.3 1221.
212-64-1058 CYNTHIA NANCY    HHHHX            F   61  495 590. 613. 2.44 2 12.2 1203.
212-66-8979 CYNTHIA SUSAN    IIIIF            F  130  529 595. 680. 2.23 2 24.4 1275.
213-64-4741 RICHARD D        IIIIH            M    7  602 680. 738. 3.13 2  1.0 1418.
213-66-8286 JANET LYNN       IIIIJ            F   10 1100 600. 560. 3.01 2   .8 1160.
215-49-7629 GEORGIA C        IIIIN            F   48  653 479. 605. 2.42 2  7.2 1084.
215-50-2026 JUDITH S         IIIIO            F  114  850 576. 660. 2.49 2 13.9 1236.
215-50-8148 JULIA ANN        IIIIQ            F    9  529 608. 662. 3.01 2  1.6 1270.
215-50-8149 JENNIFER LEE     IIIIR            F  102  529 602. 536. 2.27 2 19.1 1138.
215-64-5705 GEORGE DANIEL    IIIIT            M  137  715 595. 620. 2.26 2 19.0 1215.
216-24-6074 EVELINE F        IIIIW            F   13  158       .    .    .  2   .    0.
216-48-1892 CATHERINE E      IIIIX            F    4  195 630. 681. 3.07 2  1.7 1311.
216-66-6973 BRIAN DAVID      JJJJB            M   13   96 633. 672. 2.52 2 13.0 1305.
218-52-2650 BARBARA E        JJJJE            F   20  654 550. 540. 2.59 2  2.9 1090.
218-54-7377 STEPHEN PAUL     JJJJF            M  407  763 529. 605. 1.83 2 53.2 1134.
218-54-7431 JONATHAN L       JJJJG            M  112  572 661. 710. 2.53 2 19.4 1371.
218-62-1676 LINDA G          JJJJH            F    7  464 695. 692. 3.19 2  1.4 1387.
219-56-4398 NANCY LOUISE     JJJJJ            F    1   31 582. 700. 3.02 2  1.6 1282.
220-44-6989 GARY KIRK        JJJJN            M   62   84 600. 530. 1.56 2 73.2 1130.
220-46-4507 SUSAN LEIGH      KKKKA            F    9  383 529. 640. 2.86 2  2.2 1169.
220-54-1804 ARTHUR E         KKKKD            M   50  614 700. 729. 2.82 2  8.0 1429.
220-58-7843 JOHN PHILIP      KKKKE            M   11   49 511. 610. 2.13 2 21.4 1121.
222-40-1766 CHARLES P        MMMMB            M  255  383 512. 630. 1.75 2 66.4 1142.
222-42-1674 DAVID JOHN       MMMMD            M  105  383 600. 570. 2.06 2 27.2 1170.

THE NUMBER OF RECORDS PRINTED ON THIS REPORT IS    101
```

Example 1: Listing of approved applicants

(Page 4 of 4)

```
         LISTING OF INPUT CONTROL CARDS
DATE   011572   SAMPLE REPORT 16
SORT   17
SORT   3
SORT 4
SELECTION   CRITERIA=4
FIELD=(A,17),LIMITS=('NC','NC')
FIELD=(B,17),LIMITS=('SC','SC')
FIELD=(C,10),LIMITS=('B','B')
FIELD=(D,50),  LIMITS= ('2','2')
NOT(A OR B) AND C AND D
ACCEPTED BOARDING STUDENT NOT FROM NC OR SC
PRINT   1 2 3 16 17 10 50
END
```

```
                 SELECTIVITY INSTRUCTIONS

ACCEPTED BOARDING STUDENT NOT FROM NC OR SC

BOOLEAN EXPRESSION            NOT(A OR B) AND C AND D
     BOUNDS SET 'A' IS     STATE
          LOWER LIMIT     NC
          UPPER LIMIT     NC
     BOUNDS SET 'B' IS     STATE
          LOWER LIMIT     SC
          UPPER LIMIT     SC
     BOUNDS SET 'C' IS     BOARDING OR COMMUTING
          LOWER LIMIT     B
          UPPER LIMIT     B
     BOUNDS SET 'D' IS     BOARD DECISION
          LOWER LIMIT     2
          UPPER LIMIT     2
```

Example 2: Listing of Accepted Boarding Student not from NC or SC

(Page 1 of 4)

```
THIS REPORT WAS GENERATED ON 01/15/72.
SAMPLE REPORT 16

THE FOLLOWING ARE THE COLUMN HEADINGS AND AN EXPANDED EXPLANATION

COLUMN  FIELD NO  HEADING      EXPLANATION

   1        1     SOC SEC     SOCIAL SECURITY NUMBER
   2        2     NAME        1ST & 2ND NAME OR INITIA
   3        3     LAST NAME   LAST NAME
   4       16     CITY        CITY
   5       17     ST          STATE
   6       10     B           BOARDING OR COMMUTING
   7       50     D           BOARD DECISION

 THE NUMBER OF RECORDS PRINTED ON THIS REPORT IS     92
```

Example 2: Listing of Accepted Boarding Student not from NC or SC

(Page 2 of 4)

```
DATE 01/15/72                                               PAGE   1
ACCEPTED BOARDING STUDENT NOT FROM NC OR SC

SOC SEC     NAME          LAST NAME         CITY           ST B D

045-42-3851 SHERRI LYNN   CCCCN             NEWCANAAN      CT B 2
045-46-1466 DAVID DENIS   CCCCO             WOODBRIDGE     CT B 2
045-50-4156 CLINTON T     CCCCP             MILFORD        CT B 2
046-36-9368 SHARON LEE    CCCCQ             HARTFORD       CT B 2
048-50-0423 DAVID MICHAEL CCCCT             NEW CANAAN     CT B 2
049-46-0647 DAVID CHARLES CCCCU             FARMINGTON     CT B 2
032-42-3218 JANYCE        DDDDR             NEW CANAAN     CT B 2
212-58-3262 JAMES J       MMMMV             WASHINGTON     CT B 2
222-40-1766 CHARLES P     MMMMB             WILMINGTON     DE B 2
222-42-1674 DAVID JOHN    MMMMD             WILMINGTON     DE B 2
000-71-0075 GERHARDT M    AAAAM             PALM BEACH     FL B 2
181-38-6954 GERALD P      GGGGX             LIGHTHOUSE PT  FL B 2
215-50-2026 JUDITH S      IIIIO             WINTER PARK    FL B 2
000-71-0060 JACQUELINE    BBBBD             DECATUR        GA B 2
205-40-4132 ELEANOR K     HHHHM             ATLANTA        GA B 2
078-40-9330 DAVID PHILIP  DDDDM             DEERFIELD      IL B 2
020-36-5697 DAVID TYLER   BBBBX             STOCKBRIDGE    MA B 2
029-40-5527 JOHN THOMAS   CCCCA             WALTHAM        MA B 2
212-64-1058 CYNTHIA NANCY HHHHX             ROCKVILLE      MD B 2
212-66-8979 CYNTHIA SUSAN IIIIF             POTOMAC        MD B 2
213-64-4791 RICHARD P     IIIIH             ANNAPOLIS      MD B 2
213-66-8266 JANET LYNN    IIIIJ             BALTIMORE      MD B 2
215-50-8148 JULIA ANN     IIIIO             POTOMAC        MD B 2
215-50-8149 JENNIFER LEE  IIIIR             POTOMAC        MD B 2
215-64-5705 GEORGE DANIEL IIIIT             TEMPLE HILLS   MD B 2
216-48-1872 CATHERINE E   IIIIX             BALTIMORE      MD B 2
216-66-6973 BRIAN DAVID   JJJJB             SPARKSGLENCOE  MD B 2
218-54-7377 STEPHEN PAUL  JJJJF             BETHESDA       MD B 2
218-54-7431 JONATHAN L    JJJJG             SILVER SPRING  MD B 2
218-62-1676 LINDA G       JJJJH             BALTIMORE      MD B 2
219-56-4398 NANCY LOUISE  JJJJJ             SPARKS         MD B 2
220-44-6769 GARY KIRK     JJJJN             CHEVY CHASE    MD B 2
220-46-4407 SUSAN LEIGH   KKKKA             SALISBURY      MD B 2
220-58-1604 ARTHUR E      KKKKD             SEVERNA PK     MD B 2
220-58-7843 JOHN PHILIP   KKKKE             ROCKVILLE      MD B 2
377-38-3239 LARRY LEROY   DDDDV             RINGOES        NJ B 2
138-40-8202 EDMUND A      EEEER             FORT LEE       NJ B 2
138-40-8576 ALICE V       EEEES             CRESSKILL      NJ B 2
139-50-6512 JAY B         EEEEW             HADDONFIELD    NJ B 2
143-38-2827 MICHAEL A     FFFFB             JERSEY CITY    NJ B 2
143-48-3524 CARL JOSEPH   FFFFC             SOMERVILLE     NJ B 2
144-50-0137 ROBERT C      FFFFD             SEA GIRT       NJ B 2
146-40-7338 STEVEN E      FFFFF             MIDDLETOWN     NJ B 2
147-38-3138 TERESA LOUISE FFFFI             SCOTCH PLAINS  NJ B 2
150-44-0856 ERIC STUART   FFFFM             VERONA         NJ B 2
150-48-4372 EDWARD J      FFFFN             GILLETTE       NJ B 2
153-48-3992 MICHAEL F     FFFFS             BOUND BROOK    NJ B 2
156-44-7429 JAMES         FFFFY             BRICK TOWN     NJ B 2
156-46-7312 NANCY FORREST FFFFZ             MORRIS         NJ B 2
156-48-1492 KEVIN T       GGGGA             RIDGEWOOD      NJ B 2
157-36-7681 THOMAS WARREN GGGGB             WYCKOFF        NJ B 2
157-38-9340 WILLIAM T     GGGGC             FLORHAM PARK   NJ B 2
157-46-2491 JERRY ALAN    GGGGD             SO PLAINFIELD  NJ B 2
167-44-4806 CLINTON RAY   GGGGJ             WRIGHTSTOWN    NJ B 2
212-58-6252 THADDEUS W    HHHHW             UPR MONTCLAIR  NJ B 2
```

Example 2: Listing of Accepted Boarding Student not from NC or SC

(Page 3 of 4)

```
DATE 01/15/72                                               PAGE   2
ACCEPTED BOARDING STUDENT NOT FROM NC OR SC

SOC SEC     NAME          LAST NAME         CITY           ST B D

055-46-7858 ELIAS         DDDDB             MANHASSET      NY B 2
057-46-7122 DANIEL A      DDDDC             CATSKILL       NY B 2
064-46-9520 BRIAN MICHAEL DDDDE             BAY SHORE      NY B 2
074-42-8223 VERONICA M    DDDDH             QUEEN' VILL    NY B 2
075-48-4006 ALAN CLAY     DDDDJ             NORTHPORT      NY B 2
091-46-2031 JANE IDA      DDDDQ             FRANKLIN       NY B 2
092-44-2445 HOWARD I      DDDDS             BRONX          NY B 2
096-44-8879 GARRY W       DDDDU             KINGSTON       NY B 2
117-40-1235 PAUL J        EEEEB             HUNTINGTON     NY B 2
121-46-6362 MICHAEL HAYS  EEEEH             ISLIP          NY B 2
121-46-7863 MARY HAZEL    EEEEI             GENEVA         NY B 2
122-46-5179 MARK KEITH    EEEEJ             OLEAN          NY B 2
123-46-6858 MARILYN B     EEEEK             LIVERPOOL      NY B 2
125-44-1089 BALDWIN       EEEEL             BRONX          NY B 2
125-46-9506 EDWARD T      EEEEM             STATEN ISLAND  NY B 2
163-46-2969 ELIZABETH ANN GGGGG             HUNTINGDON     PA B 2
164-46-6045 DONALD A      GGGGH             WAYNE          PA B 2
168-42-4960 WOODROW W     GGGGK             DANVILLE       PA B 2
170-46-6099 WILLIAM L     GGGGO             PITTSBURGH     PA B 2
171-38-1225 RICHARD SCOTT GGGGP             WYOMISSING     PA B 2
176-44-6228 ROBERT ALTON  GGGGS             BETHLEHEM      PA B 2
177-46-5046 WINBORN C     GGGGT             MAPLE GLEN     PA B 2
179-42-2318 JEFFREY JAY   GGGGU             GLENSHAW       PA B 2
179-42-4812 ANDREW W      GGGGV             PITTSBURGH     PA B 2
182-46-0590 JANET MARIO   GGGGY             READING        PA B 2
190-46-2314 DANIEL G      HHHHE             BEN AVON       PA B 2
190-46-2810 WILLIAM C     HHHHF             PITTSBURGH     PA B 2
205-42-6751 JANICE LEE    HHHHP             CAMPHILL       PA B 2
210-36-0373 BLAINE JOSEPH HHHHQ             SOUTH TEMPLE   PA B 2
211-44-0794 KATHLEEN B    HHHHS             HUNTINGDON     PA B 2
211-44-1493 DONALD ALAN   HHHHT             ALLENTOWN      PA B 2
035-30-4961 TOWNES M      CCCCC             PROVIDENCE     RI B 2
000-71-0078 LAWRENCE D    BBBBP             KNOXVILLE      TN B 2
000-71-0092 ROBERT B      BBBBN             KNOXVILLE      TN B 2
215-48-7829 GEORGIA C     IIIIN             ANNANDALE      VA B 2
009-40-1507 JANE AVERY    BBBBR             RUTLAND        VT B 2
000-71-0074 JOSEPH D      AAAAL             NIGERIA           B 2


        THE NUMBER OF RECORDS PRINTED ON THIS REPORT IS      92
```

Example 2: Listing of Accepted Boarding.Student not from NC or SC

(Page 4 of 4)

```
              LISTING OF INPUT CONTROL CARDS
DATE   011572       SAMPLE 23  PROBLEM 8
SELECTION CRITERIA = 2
FIELD=(A,50),LIMITS=('2','2')
FIELD=(B,10),LIMITS=('B','B')
A AND B
ACCEPTED BOARDING APPLICANTS
FREQUENCY   42  200.   50.  13
FREQUENCY   43  200.   50.  13
SELECTION CRITERIA = 1
FIELD=(A,47),LIMITS=('0001','0100')
A
HIGH SCHOOL CLASS OF 100 OR LESS
FREQUENCY   42  200.   50.  13
FREQUENCY   43  200.   50.  13
SELECTION CRITERIA = 1
FIELD=( A,47),LIMITS=('0101','9999')
A
HIGH SCHOOL CLASS OF 101 OR OVER
FREQUENCY   42  200.   50.  13
FREQUENCY   43  200.   50.  13
END
```

Example 3: Admissions Statistics

(Page 1 of 8)

```
                    SELECTIVITY INSTRUCTIONS

ACCEPTED BOARDING APPLICANTS

BOOLEAN EXPRESSION              A AND B
     BOUNDS SET 'A' IS       BOARD DECISION
          LOWER LIMIT          2
          UPPER LIMIT          2
     BOUNDS SET 'B' IS       BOARDING OR COMMUTING
          LOWER LIMIT          B
          UPPER LIMIT          B


                         SELECTIVITY INSTRUCTIONS

ACCEPTED BOARDING APPLICANTS
HIGH SCHOOL CLASS OF 100 OR LESS

BOOLEAN EXPRESSION              A
     BOUNDS SET 'A' IS       LATEST CLASS SIZE
          LOWER LIMIT        0001
          UPPER LIMIT        0100


                         SELECTIVITY INSTRUCTIONS

ACCEPTED BOARDING APPLICANTS
HIGH SCHOOL CLASS OF 101 OR OVER

BOOLEAN EXPRESSION              A
     BOUNDS SET 'A' IS       LATEST CLASS SIZE
          LOWER LIMIT        0101
          UPPER LIMIT        9999
```

Example 3: Admissions Statistics

(Page 2 of 8)

```
FREQUENCY TABLE AND SUPPLEMENTARY STATISTICS FOR SAT-VERBAL
DATE 01/15/72      SAMPLE 23  PROBLEM 8.

ACCEPTED BOARDING APPLICANTS

     INTERVAL            FREQUENCY  CUMULATIVE  PER CENT  CUMULATIVE
                                    FREQUENCY             PER CENT

200.000 AND BELOW            0           0        0.00       0.00
200.001-  250.000            0           0        0.00       0.00
250.001-  300.000            0           0        0.00       0.00
300.001-  350.000            1           1        1.04       1.04
350.001-  400.000            6           7        6.25       7.29
400.001-  450.000            9          16        9.37      16.66
450.001-  500.000           14          30       14.58      31.25
500.001-  550.000           19          49       19.79      51.04
550.001-  600.000           15          64       15.62      66.66
600.001-  650.000           14          78       14.58      81.25
650.001-  700.000           15          93       15.62      96.87
700.001-  750.000            3          96        3.12     100.00
750.001-  800.000            0          96        0.00     100.00
800.001 AND ABOVE            0          96        0.00     100.00

SUPPLEMENTARY STATISTICS

   THE MAXIMUM VALUE WAS    733.000
   THE MINIMUM VALUE WAS    310.000
   THE AVERAGE VALUE IS     552.708
   THE STANDARD DEVIATION IS     96.378
```

Example 3: Admissions Statistics

(Page 3 of 8)

```
FREQUENCY TABLE AND SUPPLEMENTARY STATISTICS FOR SAT-MATH
DATE 01/15/72      SAMPLE 23  PROBLEM 8

ACCEPTED BOARDING APPLICANTS

     INTERVAL            FREQUENCY  CUMULATIVE  PER CENT  CUMULATIVE
                                    FREQUENCY             PER CENT

200.000 AND BELOW            0           0        0.00       0.00
200.001-  250.000            0           0        0.00       0.00
250.001-  300.000            1           1        1.04       1.04
300.001-  350.000            0           1        0.00       1.04
350.001-  400.000            3           4        3.12       4.16
400.001-  450.000            2           6        2.09       6.25
450.001-  500.000            4          10        4.16      10.41
500.001-  550.000           16          26       16.66      27.08
550.001-  600.000           20          46       20.83      47.91
600.001-  650.000           20          66       20.83      68.75
650.001-  700.000           17          83       17.70      86.45
700.001-  750.000           10          93       10.41      96.87
750.001-  800.000            3          96        3.12     100.00
800.001 AND ABOVE            0          96        0.00     100.00

SUPPLEMENTARY STATISTICS

   THE MAXIMUM VALUE WAS    777.000
   THE MINIMUM VALUE WAS    300.000
   THE AVERAGE VALUE IS     603.604
   THE STANDARD DEVIATION IS     92.646
```

Example 3: Admissions Statistics

(Page 4 of 8)

```
FREQUENCY TABLE AND SUPPLEMENTARY STATISTICS FOR SAT-VERBAL
DATE 01/15/72      SAMPLE 23  PROBLEM 8

ACCEPTED BOARDING APPLICANTS

HIGH SCHOOL CLASS OF 100 OR LESS

    INTERVAL            FREQUENCY  CUMULATIVE  PER CENT  CUMULATIVE
                                   FREQUENCY             PER CENT

 200.000 AND BELOW          0          0         0.00       0.00
 200.001-  250.000          0          0         0.00       0.00
 250.001-  300.000          0          0         0.00       0.00
 300.001-  350.000          1          1         5.00       5.00
 350.001-  400.000          2          3        10.00      15.00
 400.001-  450.000          1          4         5.00      20.00
 450.001-  500.000          5          9        25.00      45.00
 500.001-  550.000          4         13        20.00      65.00
 550.001-  600.000          3         16        15.00      80.00
 600.001-  650.000          3         19        15.00      95.00
 650.001-  700.000          1         20         5.00     100.00
 700.001-  750.000          0         20         0.00     100.00
 750.001-  800.000          0         20         0.00     100.00
 800.001 AND ABOVE          0         20         0.00     100.00

SUPPLEMENTARY STATISTICS

   THE MAXIMUM VALUE WAS    700.000
   THE MINIMUM VALUE WAS    310.000
   THE AVERAGE VALUE IS     519.150
   THE STANDARD DEVIATION IS    96.331
```

Example 3: Admissions Statistics

(Page 5 of 8)

```
FREQUENCY TABLE AND SUPPLEMENTARY STATISTICS FOR SAT-MATH
DATE 01/15/72      SAMPLE 23  PROBLEM 8

ACCEPTED BOARDING APPLICANTS

HIGH SCHOOL CLASS OF 100 OR LESS

    INTERVAL            FREQUENCY  CUMULATIVE  PER CENT  CUMULATIVE
                                   FREQUENCY             PER CENT

 200.000 AND BELOW          0          0         0.00       0.00
 200.001-  250.000          0          0         0.00       0.00
 250.001-  300.000          1          1         5.00       5.00
 300.001-  350.000          0          1         0.00       5.00
 350.001-  400.000          1          2         5.00      10.00
 400.001-  450.000          1          3         5.00      15.00
 450.001-  500.000          1          4         5.00      20.00
 500.001-  550.000          4          8        20.00      40.00
 550.001-  600.000          4         12        20.00      60.00
 600.001-  650.000          2         14        10.00      70.00
 650.001-  700.000          5         19        25.00      95.00
 700.001-  750.000          1         20         5.00     100.00
 750.001-  800.000          0         20         0.00     100.00
 800.001 AND ABOVE          0         20         0.00     100.00

SUPPLEMENTARY STATISTICS

   THE MAXIMUM VALUE WAS    730.000
   THE MINIMUM VALUE WAS    300.000
   THE AVERAGE VALUE IS     569.350
   THE STANDARD DEVIATION IS    108.681
```

Example 3: Admissions Statistics

(Page 6 of 8)

```
FREQUENCY TABLE AND SUPPLEMENTARY STATISTICS FOR SAT-VERBAL
DATE 01/15/72      SAMPLE 23  PROBLEM 8

ACCEPTED BOARDING APPLICANTS

HIGH SCHOOL CLASS OF 101 OR OVER

     INTERVAL           FREQUENCY  CUMULATIVE  PER CENT  CUMULATIVE
                                   FREQUENCY             PER CENT

 200.000 AND BELOW          0          0         0.00       0.00
 200.001-  250.000          0          0         0.00       0.00
 250.001-  300.000          0          0         0.00       0.00
 300.001-  350.000          0          0         0.00       0.00
 350.001-  400.000          4          4         5.26       5.26
 400.001-  450.000          8         12        10.52      15.78
 450.001-  500.000          9         21        11.84      27.63
 500.001-  550.000         15         36        19.73      47.36
 550.001-  600.000         12         48        15.78      63.15
 600.001-  650.000         11         59        14.47      77.63
 650.001-  700.000         14         73        18.42      96.05
 700.001-  750.000          3         76         3.94     100.00
 750.001-  800.000          0         76         0.00     100.00
 800.001 AND ABOVE          0         76         0.00     100.00

SUPPLEMENTARY STATISTICS

   THE MAXIMUM VALUE WAS    733.000
   THE MINIMUM VALUE WAS    376.000
   THE AVERAGE VALUE IS     561.539
   THE STANDARD DEVIATION IS     94.429
```

Example 3: Admissions Statistics

(Page 7 of 8)

```
FREQUENCY TABLE AND SUPPLEMENTARY STATISTICS FOR SAT-MATH
DATE 01/15/72      SAMPLE 23  PROBLEM 8

ACCEPTED BOARDING APPLICANTS

HIGH SCHOOL CLASS OF 101 OR OVER

     INTERVAL           FREQUENCY  CUMULATIVE  PER CENT  CUMULATIVE
                                   FREQUENCY             PER CENT

 200.000 AND BELOW          0          0         0.00       0.00
 200.001-  250.000          0          0         0.00       0.00
 250.001-  300.000          0          0         0.00       0.00
 300.001-  350.000          0          0         0.00       0.00
 350.001-  400.000          2          2         2.63       2.63
 400.001-  450.000          1          3         1.31       3.94
 450.001-  500.000          3          6         3.94       7.89
 500.001-  550.000         12         18        15.78      23.68
 550.001-  600.000         16         34        21.05      44.73
 600.001-  650.000         18         52        23.68      68.42
 650.001-  700.000         12         64        15.78      84.21
 700.001-  750.000          9         73        11.84      96.05
 750.001-  800.000          3         76         3.94     100.00
 800.001 AND ABOVE          0         76         0.00     100.00

SUPPLEMENTARY STATISTICS

   THE MAXIMUM VALUE WAS    777.000
   THE MINIMUM VALUE WAS    352.000
   THE AVERAGE VALUE IS     617.618
   THE STANDARD DEVIATION IS     85.696
```

Example 3: Admissions Statistics

(Page 8 of 8)

```
                LISTING OF INPUT CONTROL CARDS
DATE   011572   SAMPLE 19
SELECTION CRITERIA = 1
FIELD=(A,50),LIMITS=(' ','0')
A
ALL APPLICANTS
FREQUENCY   57  400.  100.  13
FREQUENCY  45  1.6  .2  13
FREQUENCY  54  0.  10.  11
CORRELATION  42 45
CORRELATION  43 45
CORRELATION  42 43
CORRELATION  57 46
END


                          SELECTIVITY INSTRUCTIONS

ALL APPLICANTS

BOULEAN EXPRESSION              A
    BOUNDS SET 'A' IS     BOARD DECISION
           LOWER LIMIT     0
           UPPER LIMIT
```

Example 4: Admissions Statistics

(Page 1 of 5)

```
CALCULATION OF CORRELATION COEFFICIENTS      DATE 01/15/72
SAMPLE 19

VARIABLE 1                  VARIABLE 2                NUMBER   COEFFICIENT OF
                                                     OF CASES   CORRELATION

 ALL APPLICANTS
SAT-VERBAL                  PREDICTIVE GPA               233          0.736
SAT-MATH                    PREDICTIVE GPA               233          0.714
SAT-VERBAL                  SAT-MATH                     237          0.604
HIGHEST SAT TOTAL SCORE     LATEST CLASS RANK              0          0.000


SUPPLEMENTARY STATISTICS ON ABOVE VARIABLES

VARIABLE                     AVERAGE     STANDARD    NUMBER
                                         DEVIATION  OF CASES

SAT-VERBAL                   497.654      97.352      237
PREDICTIVE GPA                 1.891       0.564      233
SAT-MATH                     542.249      99.579      237
HIGHEST SAT TOTAL SCORE     1040.295     176.118      237
LATEST CLASS RANK              0.000       0.000        0
```

Example 4: Admissions Statistics

(Page 2 of 5)

```
FREQUENCY TABLE AND SUPPLEMENTARY STATISTICS FOR HIGHEST SAT TOTAL SCORE
DATE 01/15/72     SAMPLE 19


ALL APPLICANTS

      INTERVAL          FREQUENCY  CUMULATIVE  PER CENT  CUMULATIVE
                                   FREQUENCY              PER CENT

  400.000 AND BELOW         0           0        0.00       0.00
  400.001-  500.000         0           0        0.00       0.00
  500.001-  600.000         2           2        0.84       0.84
  600.001-  700.000         5           7        2.10       2.95
  700.001-  800.000        14          21        5.90       8.86
  800.001-  900.000        32          53       13.50      22.36
  900.001- 1000.000        43          96       18.14      40.50
 1000.001- 1100.000        59         155       24.89      65.40
 1100.001- 1200.000        42         197       17.72      83.12
 1200.001- 1300.000        21         218        8.86      91.98
 1300.001- 1400.000        12         230        5.06      97.04
 1400.001- 1500.000         7         237        2.95     100.00
 1500.001- 1600.000         0         237        0.00     100.00
 1600.001 AND ABOVE         0         237        0.00     100.00


SUPPLEMENTARY STATISTICS

   THE MAXIMUM VALUE WAS  1471.000
   THE MINIMUM VALUE WAS   530.000
   THE AVERAGE VALUE IS   1040.295
   THE STANDARD DEVIATION IS   176.118
```

Example 4: Admissions Statistics

(Page 3 of 5)

```
FREQUENCY TABLE AND SUPPLEMENTARY STATISTICS FOR PREDICTIVE GPA
DATE 01/15/72     SAMPLE 19


ALL APPLICANTS

      INTERVAL          FREQUENCY  CUMULATIVE  PER CENT  CUMULATIVE
                                   FREQUENCY              PER CENT

   1.600 AND BELOW         72          72       30.90      30.90
   1.601-     1.800        40         112       17.16      48.06
   1.801-     2.000        26         138       11.15      59.22
   2.001-     2.200        31         169       13.30      72.53
   2.201-     2.400        24         193       10.30      82.83
   2.401-     2.600        17         210        7.29      90.12
   2.601-     2.800         8         218        3.43      93.56
   2.801-     3.000         4         222        1.71      95.27
   3.001-     3.200         8         230        3.43      98.71
   3.201-     3.400         1         231        0.42      99.14
   3.401-     3.600         1         232        0.42      99.57
   3.601-     3.800         1         233        0.42     100.00
   3.801-     4.000         0         233        0.00     100.00
   4.001 AND ABOVE          0         233        0.00     100.00


SUPPLEMENTARY STATISTICS

   THE MAXIMUM VALUE WAS      3.780
   THE MINIMUM VALUE WAS      0.070
   THE AVERAGE VALUE IS       1.891
   THE STANDARD DEVIATION IS      0.564
```

Example 4: Admissions Statistics

(Page 4 of 5)

```
FREQUENCY TABLE AND SUPPLEMENTARY STATISTICS FOR CONVERTED RANK IN CLASS
DATE 01/15/72      SAMPLE 19


ALL APPLICANTS

         INTERVAL           FREQUENCY  CUMULATIVE  PER CENT  CUMULATIVE
                                       FREQUENCY             PER CENT

     0.000 AND BELOW             0            0      0.00        0.00
     0.001-    10.000           44           44     18.88       18.88
    10.001-    20.000           39           83     16.73       35.62
    20.001-    30.000           34          117     14.59       50.21
    30.001-    40.000           32          149     13.73       63.94
    40.001-    50.000           27          176     11.58       75.53
    50.001-    60.000           15          191      6.43       81.97
    60.001-    70.000           22          213      9.44       91.41
    70.001-    80.000           15          228      6.43       97.85
    80.001-    90.000            3          231      1.28       99.14
    90.001-   100.000            2          233      0.85      100.00
   100.001 AND ABOVE             0          233      0.00      100.00


 SUPPLEMENTARY STATISTICS

    THE MAXIMUM VALUE WAS     97.800
    THE MINIMUM VALUE WAS      0.100
    THE AVERAGE VALUE IS .    33.582
    THE STANDARD DEVIATION IS    23.753
```

Example 4: Admissions Statistics

(Page 5 of 5)

```
                  LISTING OF INPUT CONTROL CARDS
DATE  011572     SAMPLE 24
SORT  6
SORT  10
SORT  17
TABULATE
SELECTION CRITERIA = 1
FIELD=(A,50),LIMITS=('2','2')
A
ACCEPTED APPLICANTS
END
```

```
                      SELECTIVITY INSTRUCTIONS


ACCEPTED APPLICANTS

BOOLEAN EXPRESSION             A
     BOUNDS SET 'A' IS      BOARD DECISION
          LOWER LIMIT       2
          UPPER LIMIT       2
```

Example 5: Use of Tabulate

(Page 1 of 3)

```
01/15/72

    FIELD    HEADING      DESCRIPTION

      6        S          SEX
     10        B          BOARDING OR COMMUTING
     17        ST         STATE

S B ST  LEVEL 3  LEVEL 2  LEVEL 1
         TOTAL    TOTAL    TOTAL

F B CT      3

F B FL      1

F B GA      2

F B MD      9

F B NC      2

F B NJ      3

F B NY      4

F B PA      4

F B VA      1

F B VT      1        30

F C NY      1

F C PA      1

F C SC      1         3       33

M B CT      5

M B DE      2

M B FL      2

M B IL      1

M B MA      2

M B MD      8

M B NJ     17

M B NY     11

M B PA     12
```

Example 5: Use of Tabulate

(Page 2 of 3)

```
S B ST  LEVEL 3  LEVEL 2  LEVEL 1
         TOTAL    TOTAL    TOTAL

M B RI      1

M B SC      4

M B TN      2

M B         1        68       68
```

Example 5: Use of Tabulate

(Page 3 of 3)

IMPLEMENTING A COOPERATIVE COMPUTING PLAN

David F. Nyman
Associate Director
Board of Higher Education
State of Illinois

IMPLEMENTING A COOPERATIVE COMPUTING PLAN

David F. Nyman

Several plans have been designed by state agencies and regional cooperatives to better utilize that portion of the institutional budget necessary to provide adequate computing services. The success in implementing these plans has been far from uniform. To a degree, success has depended upon prior conditions and the achievement of consensus on the need for cooperation. This presentation discussed the influence that organizational control and services rendered have had on successful implementation of cooperative computing plans.

The abstract of this paper is included for the interest of the reader. The full paper will be published by the Inter-University Communication Council, Inc. (EDUCOM) in a forthcoming book authored by Dr. Charles Mosmann. For more information, please contact Mr. Nyman.

CONNECTICUT'S INFORMATION SYSTEM PLAN

Francis J. Degnan
Director
Research & Publications
Commission for Higher Education
State of Connecticut

CONNECTICUT'S INFORMATION SYSTEM PLAN

INTRODUCTION

This study was initiated in June 1972 by Chancellor Warren G. Hill of the Commission for Higher Education (CHE) in response to recommendations of the Report of the Govenor's Task Force on Services and Expenditures and in growing recognition of higher education's need for more accurate, complete and timely information. It began almost coincidentally with the Higher Education Master Planning effort. Although it was not a part of that effort, both the CHE and the Management Policy Group, which directed preparation of the Master Plan had acknowledged that the Information System (I/S) recommended in this study is a direct and necessary supporting service to the Master Plan.

The study was conducted by a System for Higher Education in Connecticut (SHEC) Task Force, under the chairmanship of Francis J. Degnan, the CHE Director of Research and Publications. The SHEC Task Force was composed of one member from each of the State's five public units of higher education shown on Figure 1, augmented by three IBM representatives who acted as technical consultants to the study. IBM's services were donated. In addition to Dr. Degnan, the Task Force consisted of Dorothy C. Goodwin, Assistant Provost and Director of Institutional Research at the University of Connecticut; Cletus Clow, Dean of Administration, Central Connecticut State College; Kenneth Summerer, Associate Executive Officer for Development and Administration, Central Office, Regional Community Colleges; and Joseph Karpowich, Dean of Students, Norwalk State Technical College. The three representatives from IBM were George Hall, Marketing Representative, Charles Lounsbury, Systems Engineer, and Roger Kalar, Marketing Representative.

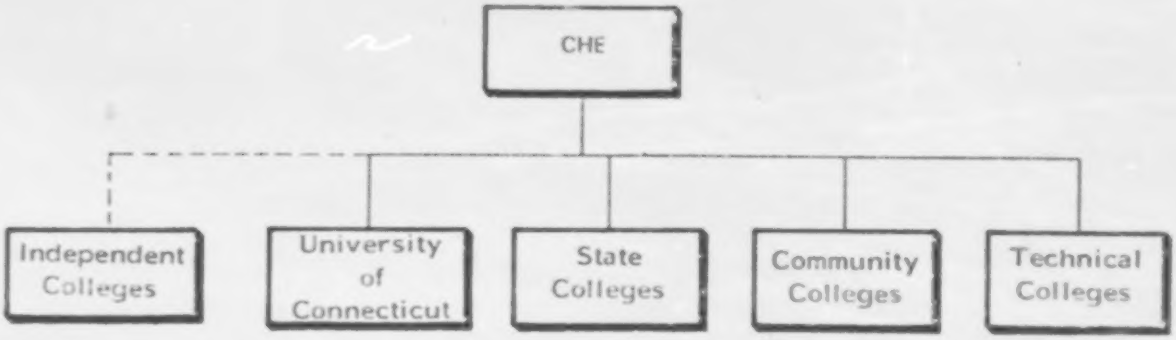


Figure 1. SHEC Organization

In its ultimate form, an I/S will be most valuable to the operation of SHEC Key executives will use it to inform judgements about such major questions as:

- How should resources be allocated to meet competing and complementary institutional objectives?
- How will changes in one area or institution affect other areas or institutions?
- How well are resources being used and objectives being met?
- How can we improve programs and ensure cost effectiveness?

Information systems do not themselves make decisions, and it is unquestionably true that the quality of decisions made on the basis of information systems will reflect the skill and judgment of the decision maker as well as the quality of the information. Nonetheless, in an environment where very complex decisions about educational/economic, or quality/cost tradeoffs must be made, a wise and capable decision maker will make better decisions with accurate, flexible and accessible information than he will without it. Executives must be able to retrieve, aggregate and analyze data in a number of different ways with "what if" capability. All parts of the system, from the individual colleges to the coordinating offices, share that need -- the need to simulate quickly the educational and financial implications of rapid shifts in policy and funding.

To meet this vital need will involve an ambitious undertaking which will take a number of years to complete. Recognizing this, the Task Force has developed a modular, building-block approach to developing the overall I/S. This approach offers three advantages:

- It yields early payback by concentrating on a first system pilot project.
- It tests concept feasibility and desirability in a manageable, small-scale environment.
- It prepares a cadre of people throughout the state who will understand the kinds of common standards and procedures that are required for such systems.

The study findings are presented in two volumes: Volume I, Executive Summary, which presents the major findings, conclusions, recommendations, actions required, and benefits, and Volume II, Study Report and I/S Plan, which presents a detailed discussion of those areas, reinforced with an Action Plan that spells out what has to be done, by whom and when. In developing its recommendations, the Task Force relied heavily on evaluations made by individual presidents and their staff members from all public institutions in Connecticut's higher educational system. The names of all those who participated actively in the study are listed in Volume II, Appendix A. To these people, and all others who worked to make the study possible, the Task Force expresses gratitude.

SCOPE AND METHOD

Scope

The study embraced all public institutions comprising Connecticut's higher education system and provided for ultimate interface with independent institutions. The SHEC Task Force, which conducted the study, had a four-part mission:

- To define an Information System (I/S) for SHEC.
- To define a management structure which would insure optimum use of the information systems resource.
- To identify and define the first or most needed system within that network.
- To develop the SHEC Information System Plan that would define the major strategies, actions, schedules and responsibilities required to achieve the I/S, its first system and a viable management plan.

It should be noted that the Task Force had a definition, not a design level mission. Its mission was to conceptualize the network, management structure and first system, and define what had to be done to get from definition to design, development, implementation and operation. The network defined in this report must be taken, one system at a time, through the design phase which will see the development of detailed specifications, including test and validation procedures.

The Nature of the I/S Concept

The key to the I/S concept is the data base - a massive compilation of non-redundant, structured, related data stored in ways that will make them instantly accessible to those who have an established need to know, in any combination of relationships the inquirer may desire.

An I/S can be developed for a single institution. If this were the objective the specifications would require simply that all parts of the institution's data base be capable of relation to all other parts. When a whole system of colleges is involved, the problem becomes much more complex since the specifications of each data element stored must be agreed upon, and that definition must be uniform throughout the system. To meet these needs there must be an agreed upon Data Element Dictionary to insure uniform use of all terms, and a Data Element Directory to show the logical arrangement of data and the records within which they are used. These two elements are generally combined into a single tool called a Data Dictionary Directory System (DD/DS).

The lowest level of information system sophistication is simple record keeping. At this level, the requirement is almost entirely a constituent unit requirement, i.e. keeping track of students, making sure the staff gets paid, entering encumbrances and rendering bills, etc. All institutions now engage in these kinds of record keeping, but much of this work is still done manually. There has been no attempt to reach agreement on common elements and definitions.

At a higher level of sophistication is the concern for operational and fiscal control. Fiscal control is the motive behind line item, character and object budget formats. It is also one of the primary emphases of the Governor's and Comptroller's offices, a concern shared by the constituent units. Today this control objective is met by agregating and rearranging the kinds of records involved in the simplest level of analysis. All institutions are currently required to maintain fiscal controls, although again, much of the work is still done manually. There is no system-wide uniformity of method or format beyond that required by the Budget Division of the Department of Finance and Control. The information generated from such a system is often inadequate and untimely. It requires considerable "translation", and, since the definition of terms and content is not uniform among constitutents, that translation involves constant rechecking and increases the chances of interpretative errors.

The ultimate objective of the I/S, however, and one still imperfectly realized virtually everywhere, is to support better management by facilitating better resource allocation. This is the level where regrouping of data instantaneously into new and flexibly-defined relationships - what has come to be called "massaging" data - is required. All parts of the system, from the individual college to the Governor's office, share this need. In the last analysis, simulation using models is the most important need; it is the need we now meet least well; it is the point at which the state of the art is least advanced.

To summarize, then, the goal is to start with data, build them into records which permit adequate fiscal control, and lead to management by objectives and assessment of planned versus actual performance, rather than by fiscal control alone.

Method

In order to develop its recommendations, the Task Force had to understand the objectives of each constituent unit, the information required to meet those objectives, how well those information needs were being met by existing systems, and what provisions were in place to measure planned versus actual progress. To gain that understanding, it divided the study into the following major steps:

- Develop a Pre-Study Document defining study objectives, scope, schedules, responsibilities, methodolgy and output.
- Review MIS state of the art in key educational and industrial projects.
- Assess information needs of constituent units.
- Develop inventory of existing hardware, personnel, budge and software within the constituent units.
- Identify common institutional processes.
- Relate data bases, processes, functions and institutions.
- Analyze findings and prepare Study Report and I/S Plan.

With regard to the second step, the Task Force studied the reports published by the Western Interstate Commission for Higher Education (WICHE) and by its division, the National Center for Higher Education Management Systems (NCHEMS).

Funded mainly from federal sources, these groups have been working toward the ultimate goal of defining an integrated I/S for all higher education in the nation. This proposal could not have been made in its present form without the work done by WICHE/NCHEMS on the selection and definition of data elements, the concepts of data bases developed by them and their studies on student flow and resource allocation models.

The steps concerning processes and data bases, etc., involve a unique approach. Traditionally, information systems studies have tended to be three dimensional. The investigator says: "I have these functions (organizations), I need a number of information systems to perform these functions, and I'll divide or catalog my data into the following data bases." There are two shortcomings to this approach: it makes information systems functionally or organizationally dependent. When the organization shifts substantially, the systems must be overhauled or scrapped; and functional/organizational systems tend to promote redundant or autonomous systems since they are built to respond to the individual functions in each organization. In recognition of these shortcomings, the Task Force introduced the concept of processes since they represent activities common, or nearly common, to all institutions. It defined processes as the activities an institution performs to achieve it objectives. In all, some 50 universal processes were identified, e.g., admissions, registration, record keeping, testing etc. The circular links between processes, systems, data bases and functions are shown in Figure 2. The matrix shows how processes interrelate with functions, which have input/output requirements for data bases, which are used by systems, which support processes.

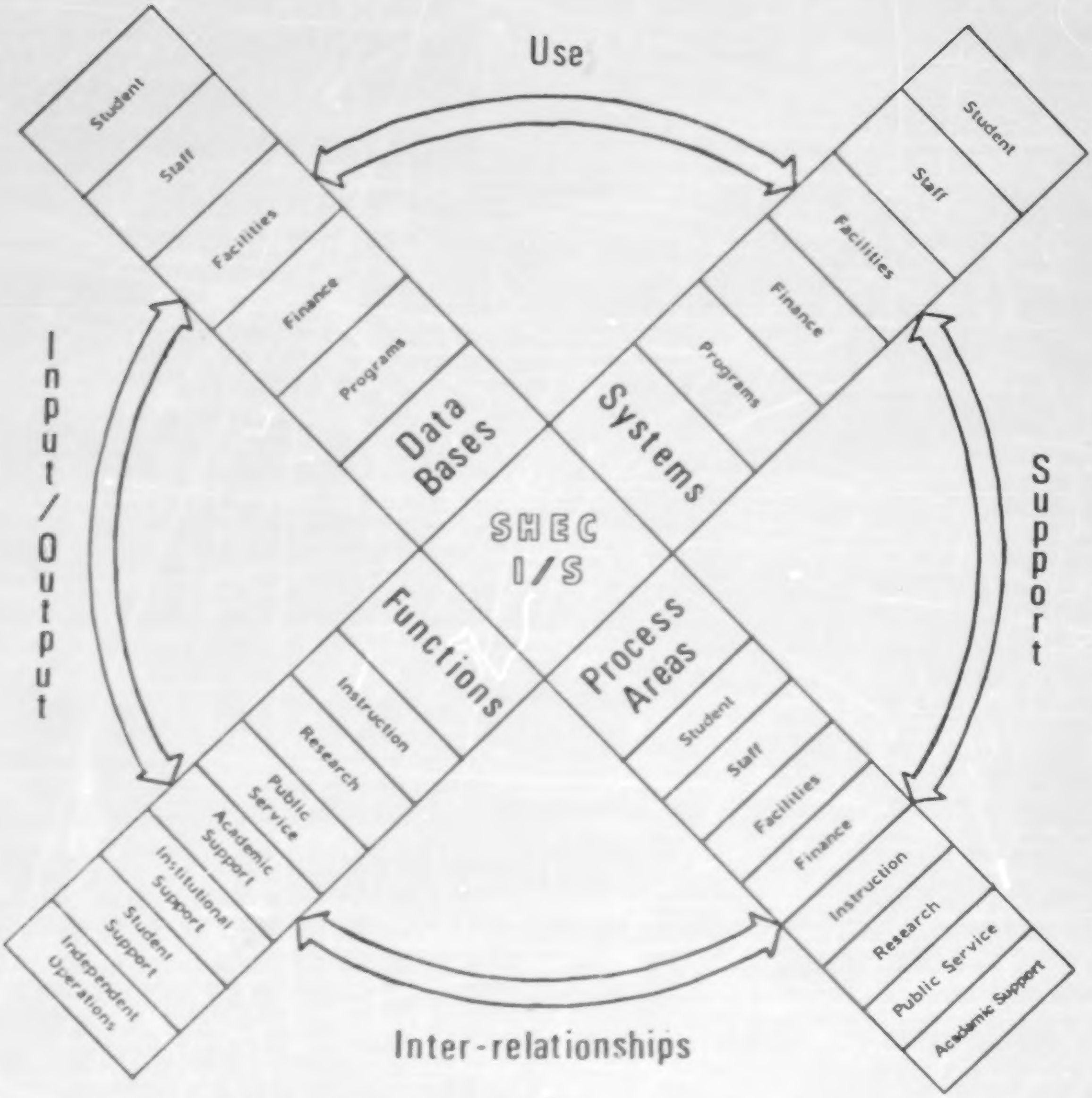

Use
Student
Staff
Facilities
Finance
Programs
Data Bases
Student
Staff
Facilities
Finance
Programs
Systems
Input/Output
SHEC I/S
Support
Functions
Instruction
Research
Public Service
Academic Support
Institutional Support
Student Support
Independent Operations
Process Areas
Student
Staff
Facilities
Finance
Instruction
Research
Public Service
Academic Support
Inter-relationships

Figure 2. Links Between Data Bases, Processes, Functions and Systems

This summary was reproduced from Volume I, Executive Summary, Study Report and Information System Plan, System for Higher Education in Connecticut, published by the Connecticut Commission for Higher Education, 340 Capitol Avenue, Hartford, Connecticut 06101. Copies of the full report may be obtained from the Commission.

A STATEWIDE APPROACH TO MANAGEMENT INFORMATION SYSTEMS FOR VOCATIONAL AND TECHNICAL INSTITUTIONS

K. M. Hussain
Professor
Computer Science
New Mexico State University

D. M. Nowrasteh
Director
Management Information
Project
Wisconsin Board of
Vocational, Technical
and Adult Education

CONTENTS

Page

1. INTRODUCTION 1

2. OVERVIEW OF THE MANAGEMENT INFORMATION SYSTEM PROJECT

2.1 Data Element Dictionaries 1

2.2 Structure of Management Information System 7

2.3 Selection of Planning and Budget Model 7

2.3.1 The CAMPUS/WISCONSIN Model 9

3. TRAINING 15

4. ROLES OF INSTITUTION AND STATEWIDE SYSTEM 15

5. PROBLEMS IN IMPLEMENTATION 15

1. INTRODUCTION

Since 1911, Wisconsin has been the only state in the nation which has maintained "separate" systems for general and occupational education. Dr. Charles McCarthy was the architect of the vocational-technical education delivery system in Wisconsin. This philosophy of vocational-technical education was termed "democratic education". Education must reach all the people, not merely those who could survive the hurdles of elementary school, or the "unit" hurdles of high school, or the "semester hour credits" of college. The concern was to serve those who dropped or were dropped out of the school system.

At present, Wisconsin's post-secondary vocational, technical and adult education programs are offered in over 600 vocational and adult centers in communities located throughout the state (see Map I). These offerings are administered through sixteen districts (see Map II). These vary in size from 600 FTE's to 9,000 FTE's.

In 1972, the Wisconsin System of Vocational, Technical and Adult Education (VTAE) served over 280,000 of the approximately 3 million Wisconsin citizens 16 years of age and over -- including 65,000 full- and part-time students in 1,000 approved post-secondary programs, and 215,000 individuals in more than 6,000 adult courses.

The Vocational, Technical and Adult Education System in FY 1974 will require over $125 million of tax resources to fund its operating expenditures.

To plan and administer such a system at the state level it is necessary to get data on each institution. Such data in the past has been inadequate in content, inconsistent in definitions, untimely in being submitted and varied greatly in accuracy and reliability. To overcome this problem, the State Board of Vocational, Technical and Adult Education appointed a Director of Management Information Systems with the charge of designing and implementing a statewide information system. This system is the subject of this paper. It will identify the progress of implementation thus far and future plans as well as discuss the structure and components of the information system, and some of its problems of implementation.

2. OVERVIEW OF THE MANAGEMENT INFORMATION SYSTEM PROJECT

The project was funded in October, 1972, and was planned in the following (see Figure 1) phases:

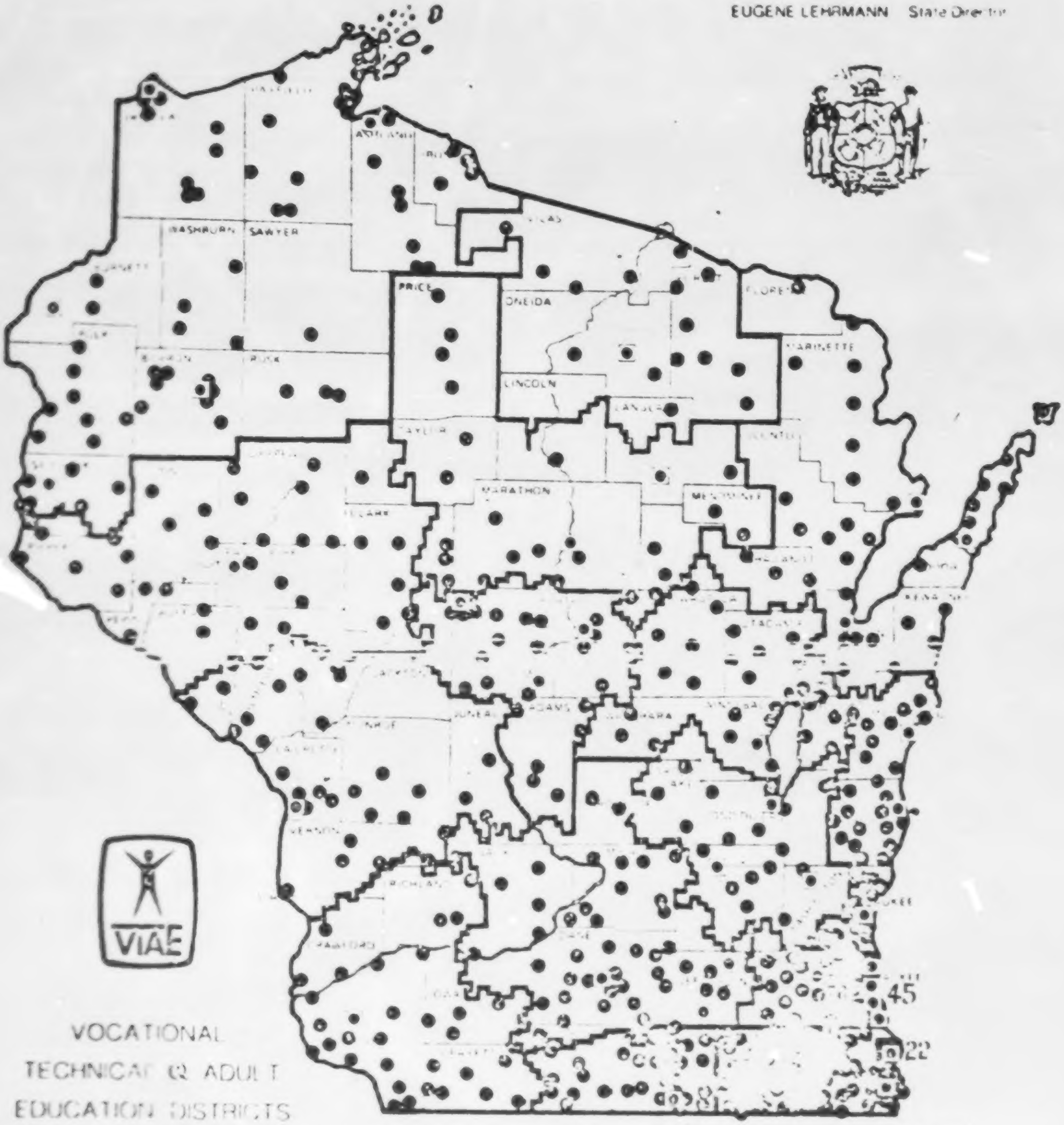
2.1 Data Element Dictionaries

The first phase of the project encompassed the development of a set of six data element dictionaries (i.e., student, staff, course, financial, facilities, socio-economic) to support the Management Information System that would generate reports required by district, regional, state and federal agencies. In each case the Management Information System staff developed a working draft of the dictionary and circulated it to each Vocational, Technical and Adult Education District and the State Board staff for modifications for subsequent drafts.

State of Wisconsin

BOARD OF VOCATIONAL, TECHNICAL & ADULT EDUCATION

EUGENE LEHRMANN State Director


BAYFIELD
ASHLAND
WASHBURN
SAWYER
BURNETT
POLK
BARRON
RUSK
PRICE
VILAS
ONEIDA
LINCOLN
LANGLADE
FLORENCE
MARINETTE
TAYLOR
MARATHON
MENOMINEE
CLARK
JUNEAU
ADAMS
RICHLAND
VERNON
45
22

VTAE

VOCATIONAL
TECHNICAL & ADULT
EDUCATION DISTRICTS
AND FULL TIME CAMPUSES
...with Adult and Outreach Centers

● Full Time Designated and Approved Programs

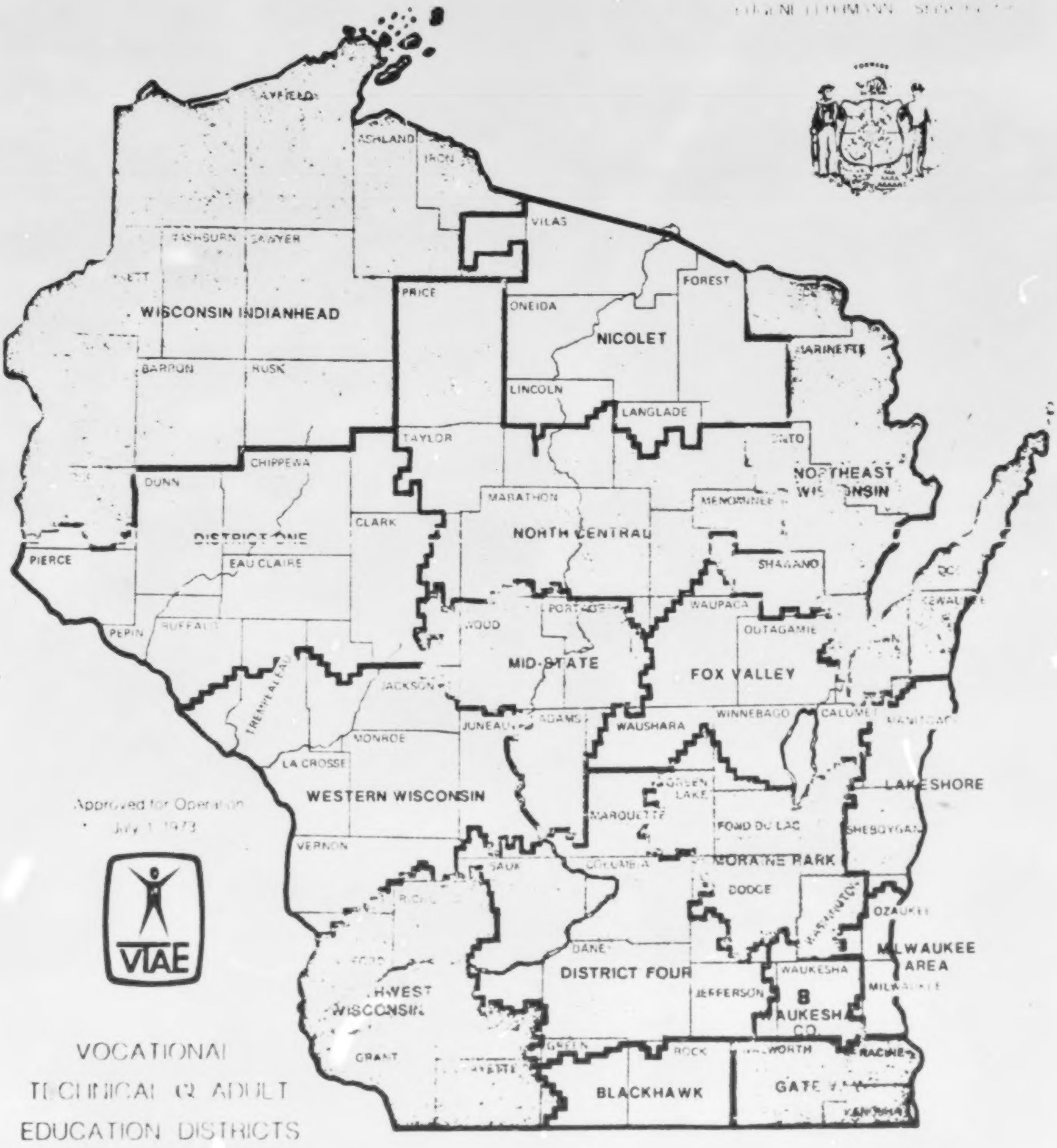
● Adult and Outreach Centers

Map I

State of Wisconsin

BOARD OF VOCATIONAL, TECHNICAL & ADULT EDUCATION

EUGENE LEHRMANN, STATE DIRECTOR


BAYFIELD
ASHLAND
IRON
VILAS
WASHBURN
SAWYER
BURNETT
WISCONSIN INDIANHEAD
PRICE
ONEIDA
FOREST
NICOLET
MARINETTE
BARRON
RUSK
LINCOLN
LANGLADE
TAYLOR
CHIPPEWA
DUNN
OCONTO
NORTHEAST WISCONSIN
MARATHON
MENOMINEE
CLARK
DISTRICT ONE
NORTH CENTRAL
PIERCE
EAU CLAIRE
SHAWANO
DOOR
PORTAGE
WAUPACA
KEWAUNEE
PEPIN
BUFFALO
WOOD
OUTAGAMIE
MID-STATE
FOX VALLEY
TREMPEALEAU
JACKSON
JUNEAU
ADAMS
WAUSHARA
WINNEBAGO
CALUMET
MANITOWOC
MONROE
LA CROSSE
GREEN LAKE
LAKESHORE
WESTERN WISCONSIN
MARQUETTE
4
FOND DU LAC
SHEBOYGAN
VERNON
SAUK
COLUMBIA
MORAINE PARK
RICHLAND
DODGE
WASHINGTON
OZAUKEE
CRAWFORD
DANE
MILWAUKEE AREA
SOUTHWEST WISCONSIN
DISTRICT FOUR
WAUKESHA
JEFFERSON
MILWAUKEE
8
WAUKESHA CO.
GRANT
IOWA
LAFAYETTE
GREEN
ROCK
WALWORTH
RACINE
BLACKHAWK
GATEWAY
KENOSHA

Approved for Operation
July 1, 1973

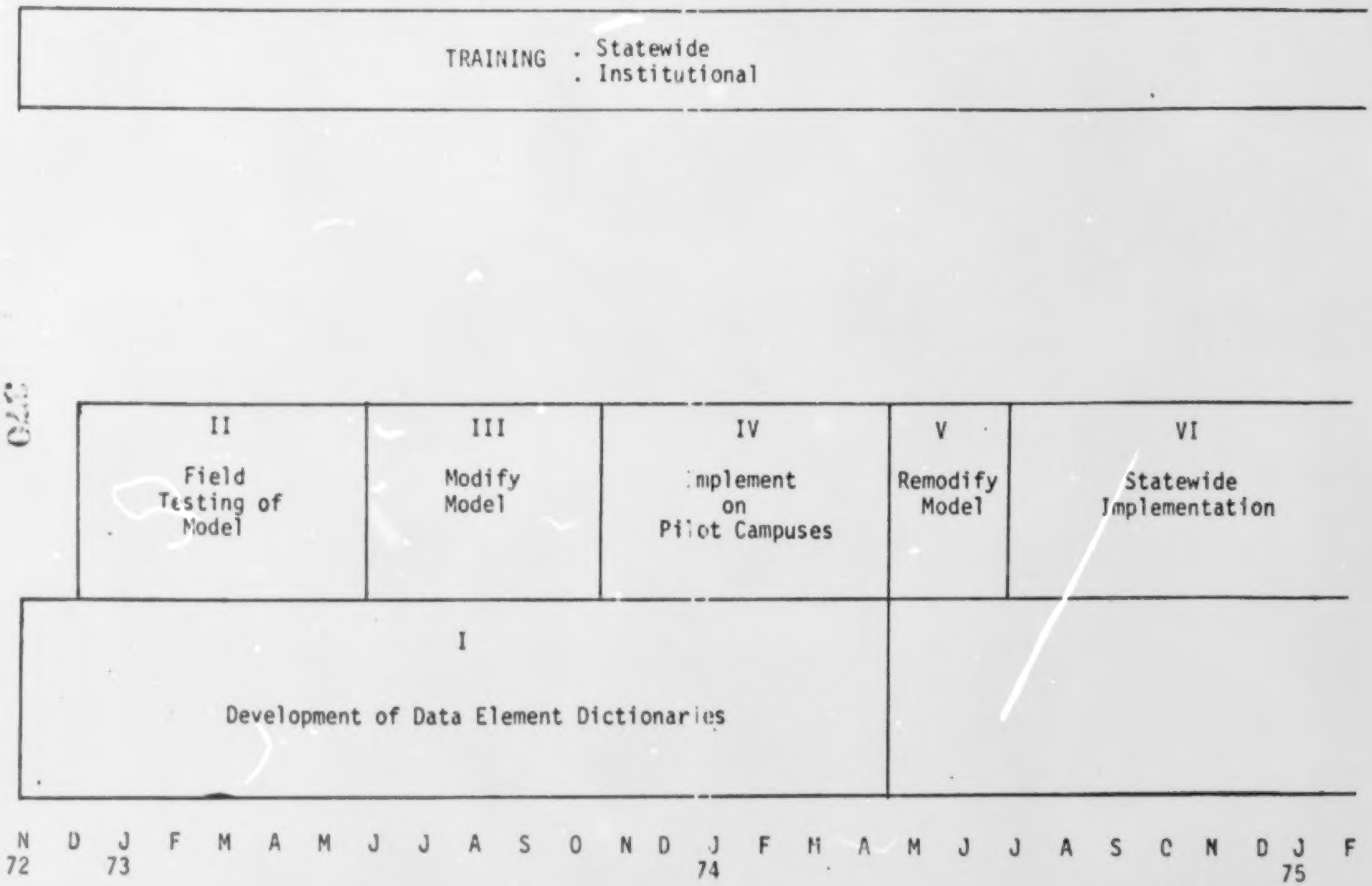
VTAE

VOCATIONAL
TECHNICAL & ADULT
EDUCATION DISTRICTS

Map II

Figure 1

Schedule for MIS Implementatior for VTAE in Wisconsin


TRAINING
. Statewide
. Institutional
II
Field Testing of Model
III
Modify Model
IV
Implement on Pilot Campuses
V
Remodify Model
VI
Statewide Implementation
I
Development of Data Element Dictionaries
N 72
D
J 73
F
M
A
M
J
J
A
S
O
N
D
J 74
F
M
A
M
J
J
A
S
O
N
D
J 75
F

A state technical advisory committee composed of each VTAE district resource persons and a State Board task force which involved State Board administrators were formed to provide inputs for second drafts of the Data Element Dictionaries, implementation of the Management Information System, and in-service training programs at the institutional and the State Board level. The Director of the Management Information System was appointed chairman of both the Technical Advisory Committee and the State Board Task Force.

Incorporation of modifications to the second draft led to the third draft. This draft of each Data Element Dictionary would then be adopted and implemented. An important factor in development of the Data Element Dictionaries was to make them mappable (wherever possible) into the Data Element Dictionaries developed by NCHEMS/WICHE. This would enable the Wisconsin VTAE System to benefit from the products of NCHEMS, especially that of cost and information exchange.

The progress of the Data Element Dictionaries is summarized in Figure 2. The first draft of the Manual on Data Element Dictionaries has been developed by the Management Information System Project staff and is now available. This document provides users with detailed specifications on data elements, as well as a summary of the data element requirements at all levels--district, state and federal.

Figure 2

Progress on the Development of the
Data Element Dictionaries

Data Element Dictionary	Draft	Distribute	Modify	Second Draft	Third Draft Adopt	Third Draft Implement
Student	X	X	X	X	X	X
Staff	X	X	X	X	X	X
Facilities	X	X	X	X	X	X
Socio-Economic	X	X	X	X	X	
Course	X	X	X	X		
Financial	X	X	X	X		

2.2 Structure of Management Information System

The second phase of the project involved determination of information needs of the State Board of VTAE. Specifically, the system was to provide information at several levels:

- Information required by the top management of the agency (i.e., State Director, assistant state directors, bureau directors, state consultants, and the State Board of VTAE)

- Generating information for production of reports as required from the agency by:

  - executive branch of the state government
  - legislative branch of the state government
  - individual departments of the state government
  - the State Vocational Education Advisory Committee
  - coordinating agencies
  - the federal government
  - other users.

The next step was to seek a software package that could be used by each institution to meet the above information needs. This relationship is shown in Figure 3.

2.3 Selection of Planning and Budgeting Model

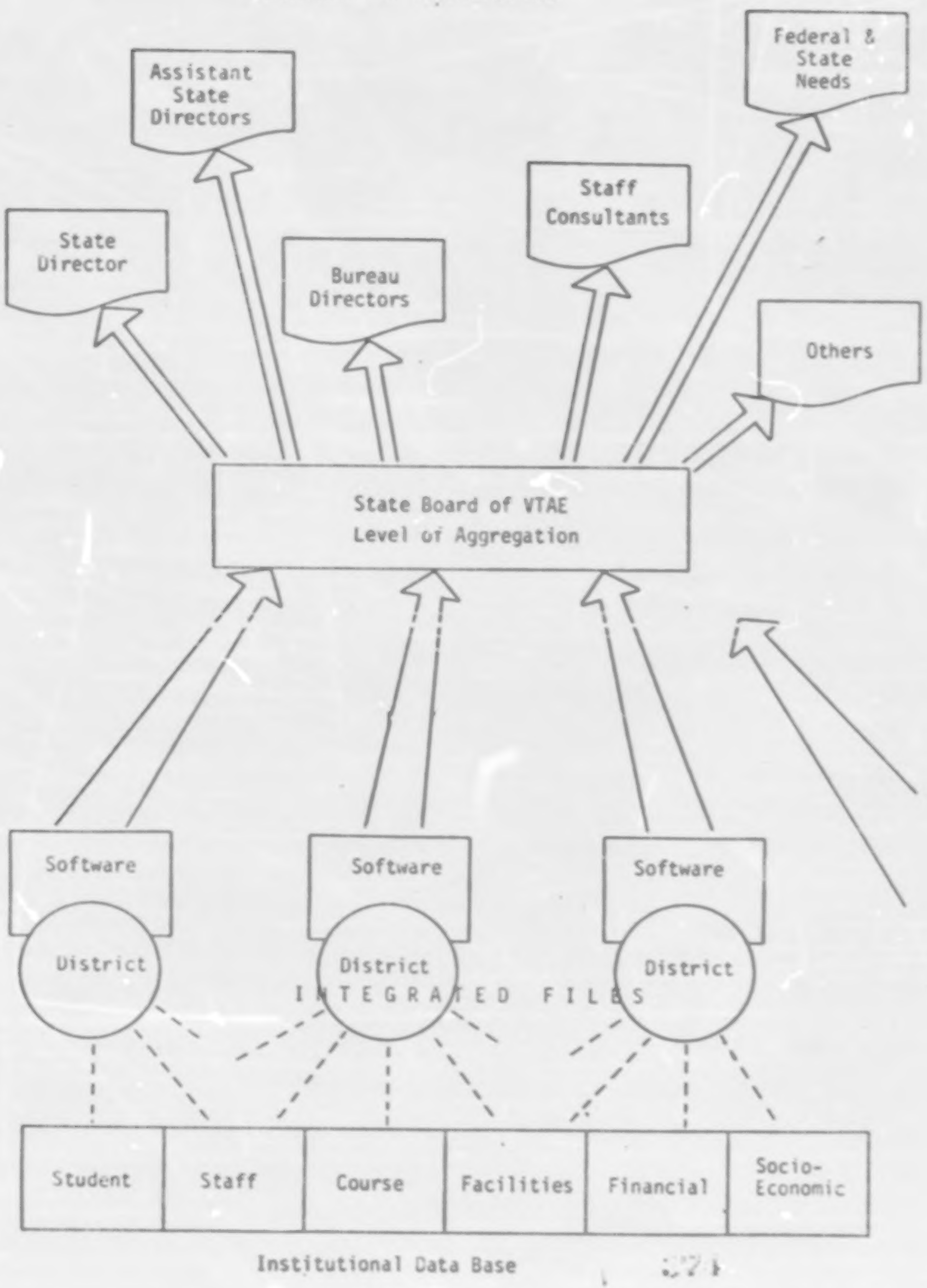
This phase of the project encompassed field testing of software packages that could meet the specific needs of occupational education delivery systems of Wisconsin. RRPM 1.6 and CAMPUS VIII models were selected. The output and the implementation experience was presented in an Information Systems Seminar held on May 10-11, 1973, for over 100 VTAE district and State Board staff to assess the feasibility of these models in meeting the unique requirements of the VTAE System.

RRPM was found to be simpler both conceptually and in its demand on input data. It is also almost a "free" good. But it did not provide the costing and planning information at the detailed activity level which is crucial for VTAE-type institutions. Furthermore, the CAMPUS model was more comprehensive; had better resolution in its computations because it did less averaging; and it had a richer set of output and a larger set of planning variables. It could meet the needs of the VTAE System provided it were modified and adapted. This SDL (the firm that markets CAMPUS) agreed to do, and was also willing to implement it on a statewide basis. Such implementation help was important to the VTAE System since it had little staff or experience with such models either at the state or institutional level.

Hence the selection of CAMPUS in its modified version, known henceforth as CAMPUS/WISCONSIN.

Figure 3

Structure of Statewide MIS


Assistant State Directors
Federal & State Needs
Staff Consultants
State Director
Bureau Directors
Others
State Board of VTAE Level of Aggregation
Software
Software
Software
District
District
District
INTEGRATED FILES
Student
Staff
Course
Facilities
Financial
Socio-Economic
Institutional Data Base

2.3.1 The CAMPUS/WISCONSIN Model

CAMPUS/WISCONSIN was programmed and three of its modules have been implemented in three institutions on a pilot basis (phase III of Figure 1). However, all the processing was done in Toronto prior to on-site implementation and maintenance at the institutions.

The programming of CAMPUS/WISCONSIN was necessitated partly because of the need to run the model on a 50K Byte machine. This is the capacity of 7 of the 12 VTAE institutions that have computers. This was possible largely through reducing the dimensions of CAMPUS VIII. A summary of the dimensions is shown in Figure 4.

The core constraint also influenced the modularity of the model. The modules of CAMPUS/WISCONSIN are shown in Figure 5. The first three modules are a subset of CAMPUS VIII. Their output is summarized in Figures 6-8. Note that the output does not provide operational reports such as class lists, grade analysis, transcripts, etc. Such output is the responsibility of each institution. With slight additions to the data base required by CAMPUS/WISCONSIN, such operational reports can be generated. Software packages are available for this purpose and are being studied by some institutions independently of the State Board of VTAE.

Figure 4

Dimensions of CAMPUS/WISCONSIN

	CAMPUS	
DIMENSION CHANGES	VIII	WISCONSIN
Course Type/Levels	8	5*
Programs	350	100
Support Space Type	110	7
Lab/Shop Type	100/50	15*
Activities	4000	2000
Faculty Ranks	10	6
Non-Academic Ranks	150	8
Other Resource Type	120	15*
Dept./Discipline	100	100
Years for Simulation	6	6
Planning Factors	114	18

*To be increased

Figure 5

Components of CAMPUS/WISCONSIN

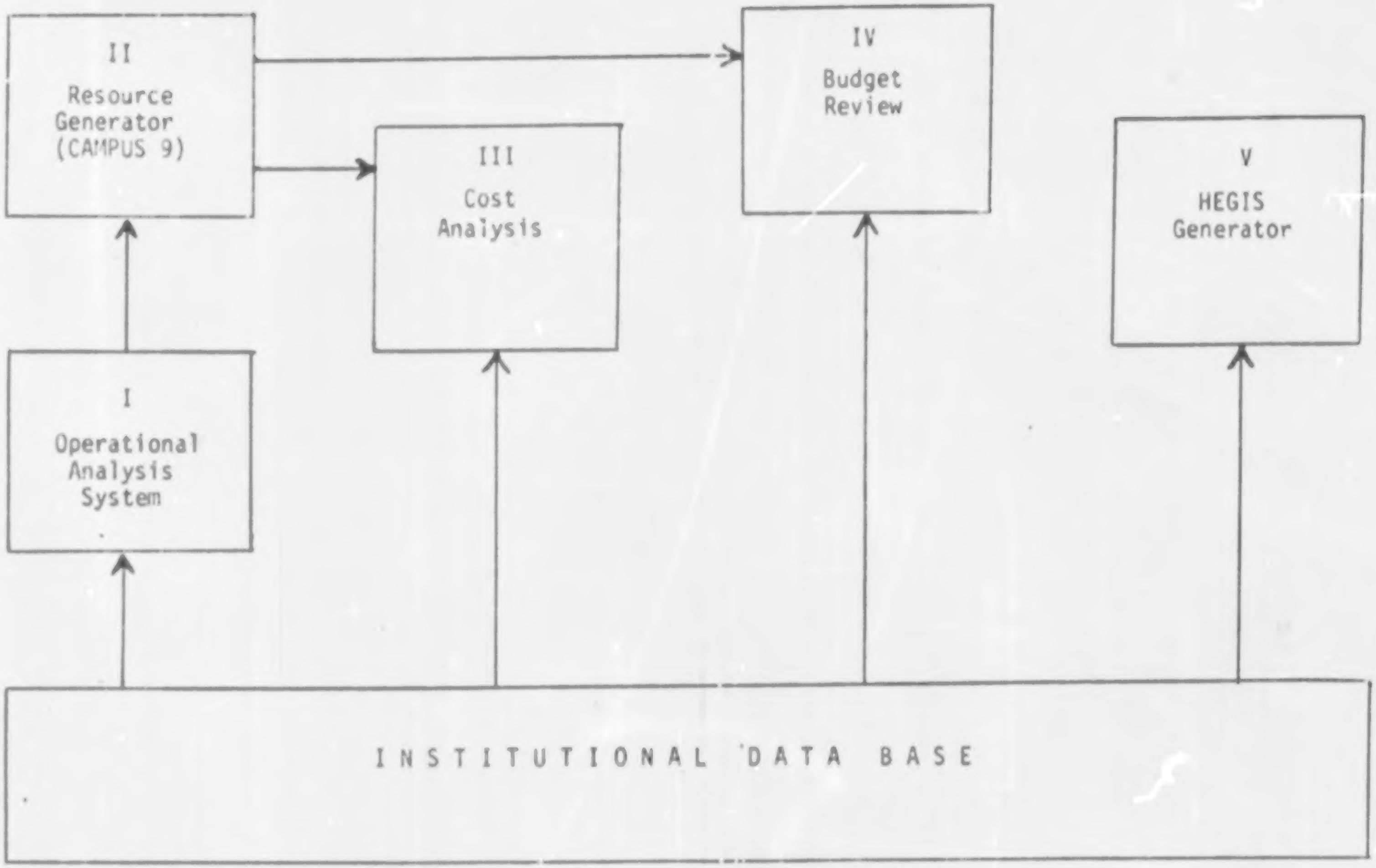

II
Resource
Generator
(CAMPUS 9)
III
Cost
Analysis
IV
Budget
Review
V
HEGIS
Generator
I
Operational
Analysis
System
INSTITUTIONAL DATA BASE

Figure 6

Output of Operations Analysis Module
Module 1

MAJOR - By Student Level

- Curriculum
- Credit Hour Load

DEPARTMENT - Loads by Major (ICLM)

DEPARTMENT OPERATING CHARACTERISTICS

- Activity Level
- Instruction Type

FACULTY SALARIES AND LOAD

- Teaching Load
- Activity Level
- Faculty Load
  - Financial Program Classification Structure (FPCS)

SPACE

- Teaching Space Type
- Non-Teaching Space Type
- Amount
- Characteristics
- Inventories
  - By Department
  - By Cost Centers

SUPPORT STAFF

- Salaries
  - By Department
  - By Cost Centers
- Fringe Benefits

INDIRECT RESOURCES

Figure 7

Basic Logic of Resource Generation, Module 2

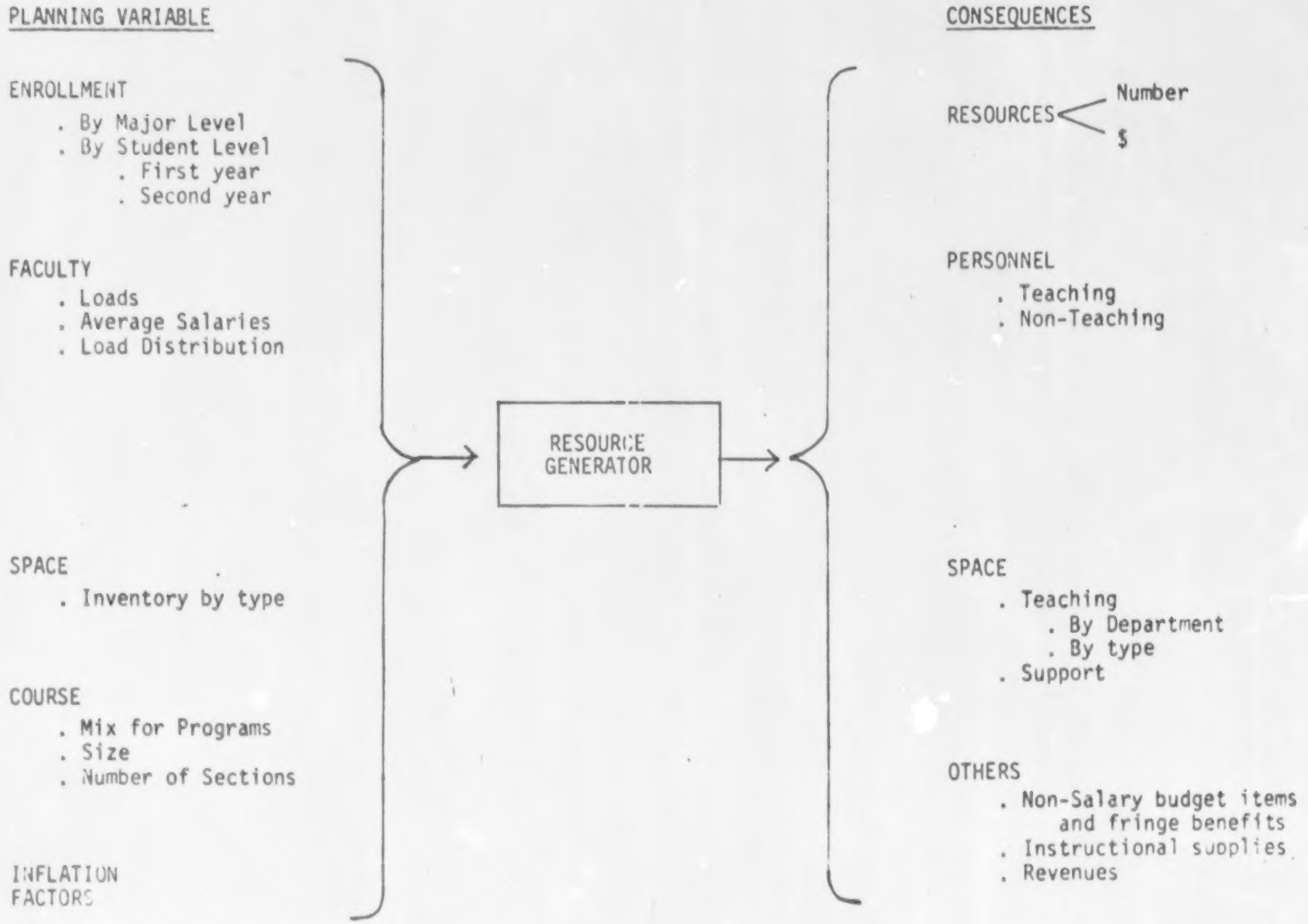
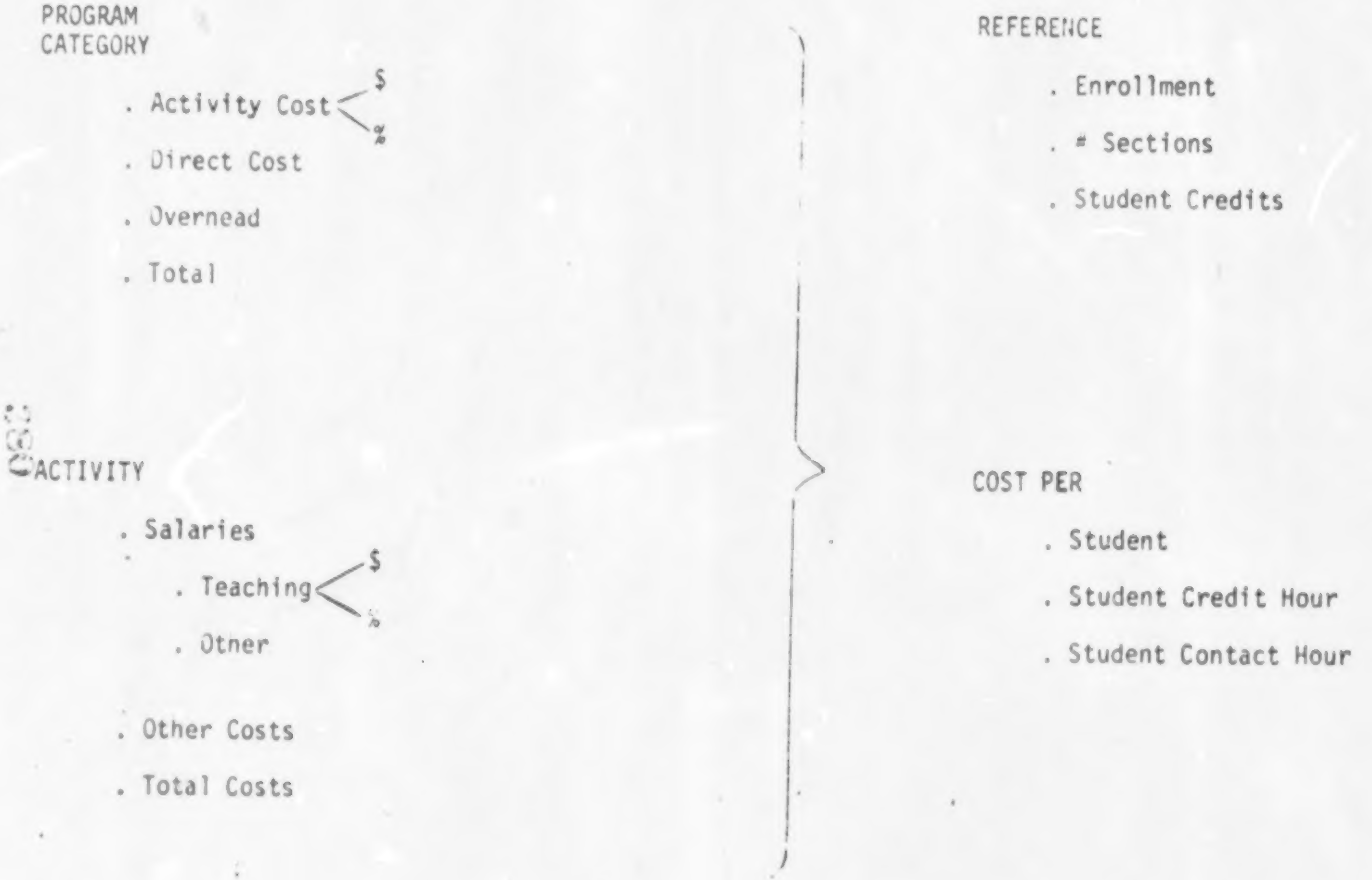

PLANNING VARIABLE
ENROLLMENT
. By Major Level
. By Student Level
. First year
. Second year
FACULTY
. Loads
. Average Salaries
. Load Distribution
SPACE
. Inventory by type
COURSE
. Mix for Programs
. Size
. Number of Sections
INFLATION FACTORS
RESOURCE GENERATOR
CONSEQUENCES
RESOURCES
Number
$
PERSONNEL
. Teaching
. Non-Teaching
SPACE
. Teaching
. By Department
. By type
. Support
OTHERS
. Non-Salary budget items and fringe benefits
. Instructional supplies
. Revenues

Figure 8

Cost Analysis, Module III



PROGRAM CATEGORY

- . Activity Cost $ %
- . Direct Cost
- . Overhead
- . Total

ACTIVITY

- . Salaries
  - . Teaching $ %
  - . Other
- . Other Costs
- . Total Costs

REFERENCE

- . Enrollment
- . # Sections
- . Student Credits

COST PER

- . Student
- . Student Credit Hour
- . Student Contact Hour

Module IV (Figure 5) has been programmed and will soon be implemented. Its output report is summarized in Figure 9. Module V will generate the HEGIS reports required by the federal government. This module is on "hold" status pending the expected changes in HEGIS reporting requirements.

The decision on when to implement CAMPUS/WISCONSIN on a statewide basis has not yet been taken (hence the open-ended part on the schedule of Figure 1). It awaits the experience and evaluation of the pilot implementation.

3. TRAINING

Parallel to all the activities discussed above, there is the activity of training (as shown in Figure 1). This is done for both state and the institution. For the institution it is done for the different levels of users: top administration, supervisory department heads, data processors and data collectors.

The media of instruction is largely through seminars and lectures. But using some modules of CAMPUS/WISCONSIN in a gaming mode is a possibility. Also a possibility, is the development of programmed instructional materials on the topic of the model.

4. ROLES OF INSTITUTION AND STATEWIDE SYSTEM

Training at the institutional level as well as the state level is the responsibility of the statewide Management Information System Project staff. They are responsible for providing the software (CAMPUS/WISCONSIN). The processing of the model and the generation of the input data is the responsibility of the institution. The reports generated will be used by the institution and the state. Their evaluation and use may then lead to modifications of the software. This feedback relationship is shown in Figure 10.

It is anticipated that a full-time person will be required at the state level to analyze the output for analytical content and to determine the needed sources of input. This analysis would help in decision making and also help in determining the future values of the planning variables. This person is shown in Figure 10, and referred to as a policy analyst. His services will also be available to the institution if so desired.

5. PROBLEMS IN IMPLEMENTATION

There are many sets of problems that are yet unresolved. One concerns the access to the output and has already raised many questions in the districts. These are: Who will get what information? And what levels of aggregation? How will it be used (or misused)?

Our approach to this problem is to bring together users from each district and put them together with their counterparts at the state level. They will discuss their uses and fears of misuse. Out of this a dialogue will emerge (hopefully) and an agreement on the access and use of each of the reports generated.

Figure 9

Budget Analyser, Module IV

- BUDGET ANALYSER
    - Function
    - Line Items
    - Cost Centers
    - Department/Divisions
    - Administrative Units

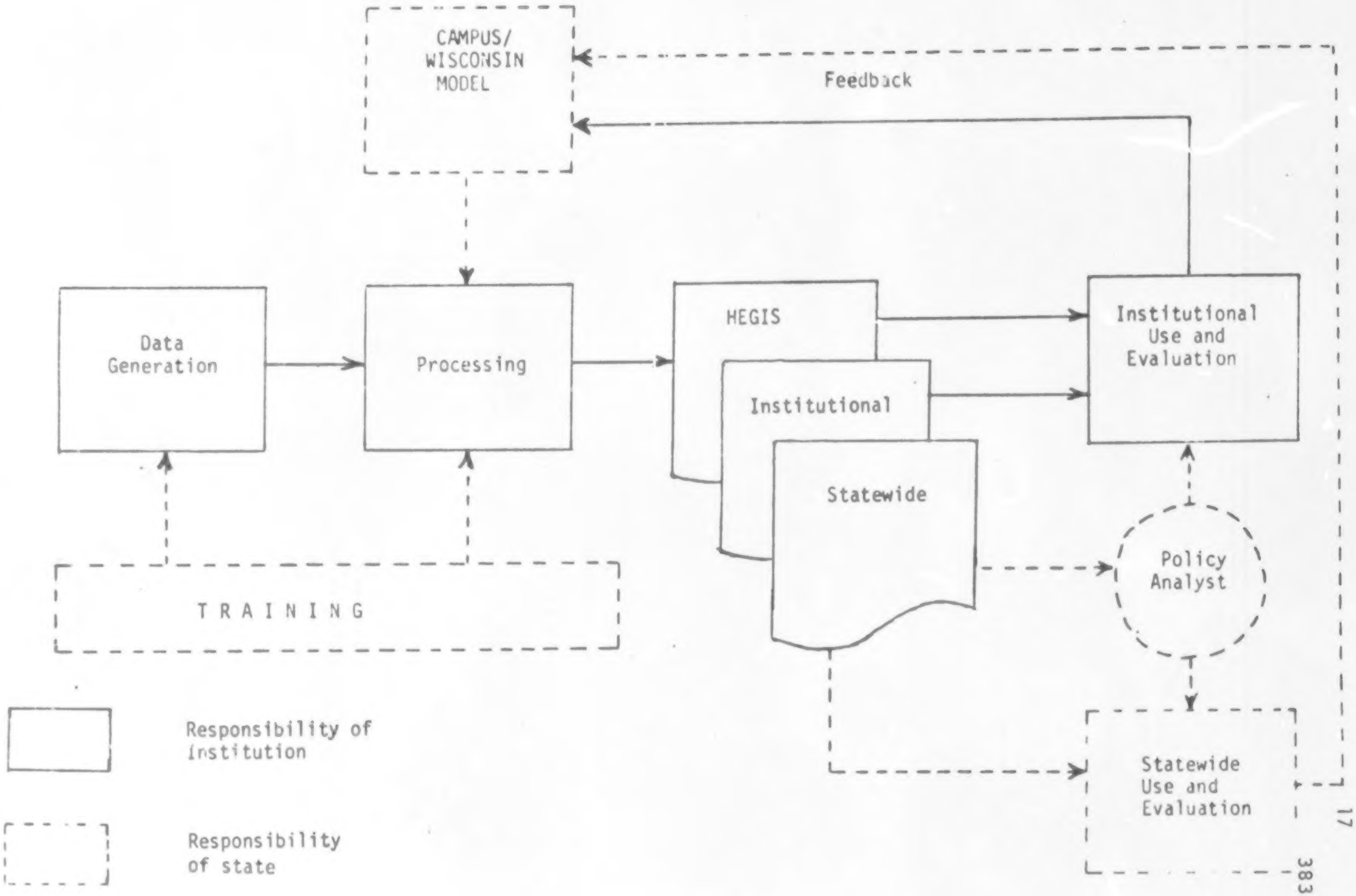
- COMPARISONS OVER TIME
    - Past
    - Current
    - Future

- SOURCE FOR DETERMINING

    Future Planning Variable

Figure 10

Roles of Institution and State


CAMPUS/
WISCONSIN
MODEL
Feedback
Data Generation
Processing
HEGIS
Institutional
Statewide
Institutional Use and Evaluation
Policy Analyst
Statewide Use and Evaluation
T R A I N I N G
Responsibility of institution
Responsibility of state

There are also hardware problems. Not all districts have the computer hardware to run the statewide software. Districts then have to form groupings for purposes of processing their data. The State Board anticipates having no hardware of its own, though it plans to have a pool of personnel that will help institutions with their processing.

The biggest challenge of the system rests in its ultimate mode of usage. The danger is that the software will be used as an accounting system rather than in answering the "what-if" type questions that arise in planning and resource allocation. If the system is so used it will be greatly under-utilized, raising the question as to its cost-effectiveness.

POINT OF SERVICE DATA RECORDING: IF IT'S HERE, WHERE IS IT GOING?

Thomas E. Doddridge
Major Accounts Representative
National Cash Register Company

POINT OF SERVICE DATA RECORDING: IF IT'S HERE, WHERE IS IT GOING?

Thomas E. Doddridge

The presentation included a brief review of turn-a-around media and its use in various environments as well as an analysis of approaches available today. Color bar code technology was discussed. Approaches in the future were suggested including magnetics and industry standardization. The systems impact of point-of-service data recording will also be covered.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

FEATURE ANALYSIS: CORE FINANCIAL SYSTEM

Dennis M. Klinger
Associate Director
Southeast Regional Data Center
Florida International University

INTRODUCTION

The Core Financial System was developed as a cooperative effort involving the Interinstitutional Committee on Finance and Accounting, the Board of Regents MIS group, and Florida International University Computer Services personnel. The purpose of the System was to develop a comprehensive Financial System capable of handling the unique fiscal requirements of the State Universities. The System was not developed for the express purpose of solving any existing emergencies; however, installing the Financial System should assist any institution currently experiencing problems in their financial transaction system.

The System is designed so that each institution will be able to implement it with as few procedural changes as possible; however, realizing the wide differences in current operational philosophy between institutions, some institutions will require more procedural changes than others.

The System references the revised edition of College and University Business Administration published by the American Council on Education in areas of principles of accounting and reporting. However, where differences exist, the Florida State University System Handbook of Financial Procedures (red book) and the Interinstitutional Committee on Finance and Accounting were the final reference.

Specific objectives of the Financial System are to provide:

1. Complete uniform accounting records for the fiscal operations of the State University System.

2. Complete and accurate information showing the current financial status of the University at the lowest operating unit on a daily basis.

3. A single source for all financial transactions data for a Management Information System at the local and Board levels, and to support planning and budgeting using PPBS.*

4. Automated by-products such as vouchers from the University's invoice processing and the supporting reports necessary for reference and audit trail.

5. Automatic generation of accounting entries from raw data which is captured from source documents whenever possible.

6. The ability and reports necessary to prepare timely annual financial statements with a minimum amount of analysis.

7. Maximum interaction with other automated systems.

8. A uniform coding scheme.

9. Options which will be flexible enough to recognize valid institutional differences but will not compromise a uniform approach.

10. A system which will provide all bookkeeping, accounting, and reporting for all functions on campus and provide for their centralization.

11. Capability for future expansion of the System as required by growth.

12. Ability to add inquiry and data collection equipment without requiring major changes in system design.

*This is to say the Core System provides a means to record and report all financial information, not necessarily generate it, i.e., payroll, receivables, etc.

DATA RECORDING

The System recognizes that the various institutions have different means of recording data in machine readable form. Consequently, the System contains no requirement for standard procedures to process the various types of transactions in preparation for the System. The System does provide several input formats. It is anticipated that forms already in use at the operating institutions might be preferred and initially adequate as input to the System. They might not, however, be compatible with the input formats. To solve this problem, it might be necessary for each institution to either modify or rewrite the first program in the System to convert their data into the format required. This would have to be done to accommodate differences in the method of input since not all universities will elect to input by cards.

CONTROLS

There are two types of error conditions that the System is designed to catch: 1) errors that are simply noted without any accompanying action by the System, and 2) errors in the input data which cause individual transactions to be rejected (not processed) by the System. Edits are provided which are common to all institutions. Provision is made for edits to be added by each institution to suit its own requirements.

Each institution has its own method of being assured that all data submitted for processing has indeed been processed and done so accurately. The System does several things to provide this assurance:

1. The System provides daily proof listings by document type and fund with appropriate totals to be used for balancing.

2. Edits are performed to eliminate as many coding errors as possible by rejecting the bad transaction before it is processed.

3. Exception messages are produced for certain conditons that are out of the ordinary. For example, journal voucher entries are checked to see that the debits equal the credits. If they do not, the condition is noted on the journal voucher proof list.

4. Programs are run which perform certain reconciliations such as matching the Budgetary and General Ledgers.

There is no procedure built into the operation of the System on the computer that requires the data processing staff to balance to control totals before allowing data to be processed. It is assumed that out-of-balance conditions will be caught whenever possible by the System, noted, and corrected on a subsequent run by those responsible for the data. It is the responsibility of the System to catch what it can, and provide enough totals to allow balancing afterward so that remaining problems can be caught on post-audit. These balancing totals, combined with adequate control procedures in the data processing department, are the means of insuring the integrity of the System reports.

ERROR CORRECTION

Corrections are made by the method of reversal and reentry. This is defined to mean that no corrections are made by deleting error records from the files and that a clear audit trail is preserved. An input error that has been exploded, thereby generating multiple errors, is corrected by reversal of the original input error and reentry of the corrected input, rather than individual correction of the exploded errors.

Correction of errors in data affecting other systems need not be corrected through the Financial System unless the Financial System is affected. For example, a change in the anticipated delivery date for a purchase order item has no effect on the Financial System even though this data is carried in one of its files. Therefore, this type of change need not be processed by the System.

GENERATION OF ACCOUNTING ENTRIES BY MACHINE FROM SOURCE DATA

The heart of the Financial System will be the point at which raw data is exploded into accounting entries. This will be accomplished, for the most part, by programming and will not require that an accountant create entries from source data or adding machine tapes. To convert the data to accounting entries, programs have been written to work on data which is of four basic types: receipts, encumbrances, disbursements, and journal entries. All input data will be handled as variations of the four basic types.

Except for journal entries, data will always be entered in the form of the transaction, not as an accounting entry. For example, a cash receipt will be entered with a single amount with appropriate coding to identify the transaction, and it will be the job of the machine to create appropriate revenue, and clearing fund entries for all ledgers from the one item. As previously stated, journal entries will be the only exception. One line on a journal entry form will result in one accounting entry in the appropriate ledger. There will be some cases in which journal entry input is appropriate but will be machine generated. One example of this is the periodic transmittal of cash to the State Treasurer and the clearing fund entries required to show this transmittal. An automatic journal entry will be prepared from clearing fund balances in the accounting records.

The entries created will be of two types--General Ledger and Budgetary or Departmental Ledger. The exact entries to be created for each type of transaction are shown in Appendix B. The entries created for the Departmental Ledger will be of a detailed nature, often down to one or more entries per line item transaction. The reason for this amount of detail is so that this file will be the single source of historical financial data for all reporting except for the General Ledger. The responsibility for summarizing the information into more useful form will fall, therefore, on the programs used to extract specialized information. The General Ledger entries, however, will be in more summarized form and will be a single entry for each document type for each fund. A lesser degree of General Ledger summary can be elected at option. This option will provide entries for each document number within type and fund.

ACCRUAL REPORTING

Since ledgers are maintained on a cash basis, it is necessary to make journal entries at period end to record accruals. Entries will be made to reflect the accrued assets and liabilities which are material in amount. These include such things as salaries payable, vendor accounts payable, accrued leave time, student receivables, and grants and contracts receivable.

Most accrual information will be readily available from other systems. Student receivables would, for example, be available from its own system. Accrued leave time information should come from the Payroll/Personnel System at period end for journalizing. The Core Financial System includes the capability for regularly reporting purchases received but not yet vouchered. Because of this option, journalizing of accounts payable at period end is automatic. If this option is not used, a manual analysis of outstanding purchase orders to determine goods received must be performed, and a journal entry made.

PPBS

The Planning-Programming-Budgeting System will be treated as another system requiring interface to the Financial System. It appears that the System coding scheme will satisfy all PPBS data requirements of the Financial System with the exception of state funded research (research positions previously generated by formula). This data can be generated elsewhere very easily.

The State University System has adopted the program coding taxonomy developed by WICHE. It is currently assumed that departments will not be engaged in more than one program. It will not, therefore, be required to code transactions to identify program.

WESTERN INTERSTATE COMMISSION FOR HIGHER EDUCATION

The State University System of Florida will participate in WICHE's Management Information Systems program at Level IV.

As previously stated, the program coding taxonomy developed by WICHE has been adopted by the Board of Regents PPBS group. Providing for PPBS coding and incorporating all eight data elements in their preliminary data dictionary (finance) insures WICHE compatibility.

LINE ITEM BASIS ENCUMBERING

The Financial System calls for a file to be maintained containing all outstanding encumbrances. Information is carried in the file as one record per purchase order line item with only a few exceptions.

The primary reason for maintaining the file is to have access to the information that was recorded at the time the purchase order was encumbered. Data such as purchase order number, amount, quantity, fund, object code, program, department, and vendor are available and required at the time that the purchase order is created. All of these items are also required at the time of payment and need not be recorded twice. Only information not available at the time of the encumbrance, such as invoice number and date or changes to the original information need be recorded.

The same reason applies in providing the ability for receiving reports to be automatically produced with an absolute minimum of information being required.

There are other benefits to maintaining a purchase order file. One of these is the ability to periodically provide the university departments with a list of all outstanding orders. This allows for a check between the Purchasing Department and the ordering department on what each believes to be on order.

Another benefit is the relative ease of producing a list of purchase orders to be certified forward at fiscal year end. This same run could, if receiving information were being carried on the file, produce the amount of accounts payable for items received but not yet vouchered.

Other systems can easily be designed to receive input from the purchase order file. Some of these will require that certain purchase orders be carried on the file on a line item basis. An example of this is a property system. Skeletal information can be saved from the purchase order file for all items **with** certain OCO object codes at the time that the item is vouchered. Then, after the item has been tagged, appropriate additional information is added to that already saved such as serial numbers and location and the item is added to the master property file.

Another application requiring line item encumbering is any automated inventory application. Automatic entry into plant, bookstore, and/or working capital inventories could be achieved at the time of vouchering.

Requirements of PPBS may reinforce the decision to encumber by-line item. The cost of goods and services for many of the programs, as they have been defined, can be provided only by being able to accurately categorize the items purchased.

One of the features of the System is a standardized set of object codes which are a refinement of the State Comptroller's object codes. Use of this object coding is designed to give more information on the way the universities' budget was spent. To be able to classify expenditures as finely as this requires that each line item be recorded separately.

SYSTEMS INTERFACE WITH THE CORE FINANCIAL SYSTEM

There will be several systems within the universities with which the Financial System must interface. This section deals with each interface and, in doing so, should show those things which the Financial System does not provide. For example, there is a very definite interface between the Payroll/Personnel System and the Financial System, but the System does not perform the functions of a Payroll/Personnel System. This is also true of the other systems whose interfaces will be described.

At this writing, the State University System has undertaken the development of several other standard systems. It is anticipated that most of the other systems will be developed for University Systemwide use. These other systems will, therefore, be required to interface with the standard systems that have preceded them in whatever ways prescribed by the preceding system. This should place no unreasonable burden upon the systems yet to be developed.

STUDENT RECEIVABLES

Student receivables transactions, as they are defined here, are all monetary transactions which occur between the university and a student with the exception of loan or scholarship transactions. Examples of transactions which should be reflected in a student receivables system are the charges and payments for library fines, housing, food service, registration fees, parking fines, breakage, etc.

It is anticipated that the Student Receivables System will receive a significant amount of its input automatically from other systems. An automated library circulation system, for example, should input charges for lost or late books to the Student Receivables System. The Student Records System should input registration fees automatically. Other types of charges will not be part of an automatic system, such as traffic fines or breakage, and will be entered manually.

The primary link between the Financial System and the Student Receivables System will be the transfer of payment information. Assuming primary input of revenue data is through the Financial System with the processing of cash receipts, most data, with some exceptions, will be transferred from the Financial System to the Student Receivables System rather than the reverse.

The data which follows this path is characterized by transactions for which a prenumbered receipt is given by the cashier's office. One exception to this flow of data is the recording of registration fees. The universities which bill their students after registration will carry this receivable in their Student Receivables System. When payment is made by the student, this data is processed by the Student Receivables System and then passed to the Financial System to record the revenue.

FINANCIAL AID

It is envisioned that an automated system for handling student financial aid will be designed as a separate system apart from ordinary student receivables. The reason for this is that except for the fact that both systems are concerned with money owed by a student to the university, they are very much different from that point on. For example:

1. Records are maintained by different departments.
2. Reporting requirements are greater for financial aid.
3. Repayment methods are very different.
4. Action for the two types is different; receivables can cause grades to be held, financial aid cannot.
5. Receivables will normally be invoiced on a regular basis, loans will not.

There are many more differences which all tend to make separate systems desirable.

There are at least two possible interface points between the Financial Aid System and the Financial System. The first is automatic vouchering from input received after checks are created each term by the computer in the Financial Aid System. The second is payment information passed to the Financial Aid System from the Financial System in its routine processing of cash receipts.

The term "Financial Aid", as it is used here, encompasses all monies passed on to students to further their education. Therefore, both loans and scholarship accounting would be affected by the input/output interface.

PERSONNEL/PAYROLL SYSTEM

The Personnel/Payroll System will provide automatic input to the Financial System to record payroll disbursements. The following is presented as one possible means of encumbering and recording salary payments in the Financial System.

At the beginning of the fiscal year, an encumbrance can be recorded for each department in the amount of the total of the salaries for the current employees, plus a percentage of this amount for salary increases that will occur during the year. This information will come from Schedule III of the operating budget. The Personnel/Payroll System will generate input to the Financial System for the following transactions:

1. Personnel appointment: An encumbrance in the amount of the total salary the employee will be paid through the end of the fiscal year plus the social security and retirement matching amount for the same period.
2. Termination: An encumbrance liquidation for the total salary amount that would have been paid from termination date through the end of the fiscal year had the employee not terminated.

3. Salary increase: An encumbrance for the increase amount times the number of pay periods remaining through the end of the fiscal year.

4. Encumbrance clearances: Each pay period an entry liquidating the salary encumbrances for that period.

5. Payments of salaries: Entries recording salary expenditures summarized by department.

PROPERTY SYSTEMS

The Financial System will provide the Property System with a record of certain capital assets purchased containing the following information: cost, description, purchasing department, object code, fund, purchase order number, voucher number, and vendor. This information is captured from the requisition or purchase order. The Financial System generates this output for any purchase with certain OCO object codes and with a cost above a designated amount. These records are edited by the property department and changes, additions, or deletions made before the records pass into the property master file.

WORKING CAPITAL AND AUXILIARIES

The Financial System is designed to keep the accounting records for all segments of the university and this includes Auxiliary and Working Capital funds. There are additional requirements of these funds beyond the scope of the Financial System, but they do lend themselves to a support interface with the Financial System. For example, the Financial System, since it can encumber on a line item basis, can produce input to and accept input from any number of inventory systems. Bookstores, plant operations, and working capital operations all have significant inventories which can benefit from automation. All that is necessary to accomplish the interface is for the System to accept input to print purchase orders with appropriate inventory codes, and to notify the inventory system when the goods are received. Programs have not been incorporated into the Financial System initially to accommodate this interface, but provision has been made to handle inventory codes.

The above mentioned systems are only a few of the interfaces the Core System envisions. We feel certain that the Core Financial System provides for all necessary interfaces in a large scale university environment. However, users must realize that the System does not perform the functions of any of the assumed interfaces.

ADMINS: DATA MANAGEMENT ON A MINI-COMPUTER

David Griffel Stuart McIntosh

ADMINS Project
Massachusetts Institute of Technology

A powerful and easy-to-use data management system has been implemented on a minicomputer* for the managing of multi-million byte complex related data bases stored on disks paks. Managing many files with complex data descriptions and tens of thousands of item lines of actual data occupying tens of millions of bytes of disk storage is completely realistic on a minicomputer. We have developed the logical specifications that could be implemented on any medium or large scale computer, on some minicomputers, and is currently operational on the SEL 7200 minicomputer. Although a secondary objective in our research, we nevertheless consider the development of an applications building environment-- extended algol-like systems language, Cobol, file handling including index sequential and sort, many utilities, flexible execution-- on inexpensive hardware both a significant and highly practical tool in the state of the art of developing inexpensive systems.

Data management in ADMINS employs concepts found in large computer data base software as well as many original concepts. However ADMINS is particularly suited to a minicomputer environment because ADMINS is very good at partitioning seemingly "large" data files according to the logic of hierarchecal coding schemes (e.g. the Standard Industrial Classification, e.g. commodity classifications) and the logic of the use of the data. ADMINS is designed to remove redundancy-- of descriptors (for the same code), of instructions (for the same operation) and of storage (for the same data code). Descriptor lists exist once, even though they describe characteristics in several files. Instructions exist once, even though they operate on data at different levels (e.g. object data, system log). The storage structures assimilate all the structure present in hierarchecal coding schemes and optimize the actual storage of data, removing storage duplications as they occur.

The main ADMINS concepts are:

(1) A Disk based storage structure developed to replace the traditional "logical record" concept used by Cobol and "ISAM" type packages for disk storage of records. Using the specially developed storage structure and access routines ADMINS is efficiently able to store multiple as well as one value per item, long binary strings and short binary strings as well as single word integers, repetitive patterns of values, text descriptors as well as numerical codes, hierarchical as well as single level codes, relations as well as attributes, and data fragments with missing items and missing categories as well as complete data sets.

---

*The ADMINS Project at MIT directed by Stuart McIntosh and David Griffel is funded under NSF Grant GJ929. The ADMINS research installation consists of a SEL 7200 virtual memory computer with disk paks.

(2) A data description language for the definition of the relational data structures of many files of related information such as descriptor records, including relational descriptors and facetted hierarchical descriptors, and data records such as transaction files, history files and report files. The relational record file norms are used by the software to organize the storage of the data on the disk in the most efficient manner. The system log contains system data and is managed as part of the data base. The system data contains the relational norms, the category norms--including grouping, facetting, and dependencies-- and the data type and picture representation information. This supplements the more common file name and location information found in conventional file systems. The file structure used in ADMINS supports both sequential and non-sequential access methods both via user written programs or via user given command level instructions. The file system is designed on a large scale, that is, for many files (relational records) and for large individual files.

(3) The use of a single data structure --the relational record-- throughout the system. A relational record is a file structure where both the data values permitted in the fields of each record and the named relationships between these data values are under complete system control. The types of values permitted include both variable length character strings, and hierarchically defined category structures including facetted classification schemes with dependencies in the coding. The file system for relational records is then used to store actual data, data descriptors (e.g. a list of occupations), aggregate data and derived reports, data sets relating data descriptors (e.g. a file of subassemblies, e.g. a decision table type recode structure), and the file directories and system log itself.

(4) A command level language for operations that work on whole files at a time to produce reports or derived files for display or further command level manipulation. The command level language approach minimizes the need for expensive ad hoc programming to produce particular data derivations and reports. The instruction language is concerned with selection, aggregation and reorganization of item lines in individual relational records and with relations logic operations working on more than one relational record at a time.

(5) A Cobol translator that allows COBOL programs to be run on data in ADMINS. The data division of the COBOL program is that part of a Cobol source program that relates the internal algorithms contained in the program to the external formats of the files to be created and accessed. The COBOL translator checks and merges the data division with the file norms and data description lists maintained in the data base, updating the system directories from the data division when necessary.

The Data Management Approach

In traditional non-data management approaches the meanings and formats of fields in records are kept in application programs (e.g. the data division in a Cobol program), in programmers heads, or in hard copy documentation. In modern data management approaches the meanings and formats of data fields are kept in the data base and not in application programs. ADMINS improves on the conventional data management approach in 4 ways.

(1) The meanings and formats of the fields are kept in the data base, including complex descriptor lists (e.g. a list of occupation descriptors) and are accessible to a set of general purpose command level programs that can perform all the standard data manipulations. The command level programs are data independent because they use the data base to ascertain the meanings and formats of the fields to be manipulated. Both descriptor lists and derived data (reports) are kept as files in the data base, in the same data structures as the data itself.

(2) not only are the fields in files named, but the relationships between fields are also named. This means, for example, the system knows the difference between a record of "person, occupation" where the relationship is "has" or "desires" or "left." Relationships in ADMINS may be considerably more complex than in the example and may involve more than two fields. Named fields in named relationship applied to cases are what we are calling relational records.

(3) Not only are the meanings, formats and relationships kept in the data base, but they are kept in the same file structure (the relational record) as the data and data descriptions being managed, where they comprise part of the system log.

(4) One set of command level operations are used on data, data description and the system log to operate both on source data and derived data (reports). Operations include a variety of item selection and aggregation functions, file reorganization, relations logic and many forms of data display and printout.

Disk Based Storage Structures For Relational Data Bases

In ADMINS a file (relational record) consists of item lines. An item line in ADMINS is like the conventional logical record concept in that all item lines in a file have the same form. However the ADMINS item line differs from the conventional logical record concept because the same item identity may be recorded in none, one, several or many item lines. For example, in a file where the data is of the form "person travels to country on date" there may be several item lines about the same person; and there may be none for some people, who have data in other files. In general relational data bases exhibit the property that most item identities require several or many item lines.

Each item line consists of data values arranged in fixed fields or positions, except for character strings which may be of variable length. The implementation of the disk based storage structure is as a sequence of ADMINS pages, controlled via a page table. (There are several canonical page sizes available.) The design of the internal format of the pages allows the item lines to be accessed randomly, to be considered virtual data spaces, to be content addressed because the page table serves also as an index to item lines maintained in sort, and to be updated without file recopying because pages contain "available space" and are capable of dividing into two on imminent overflow. The disk based storage structures are accessed via a subroutine interface which contains a full range of data manipulation calls: find, change, delete, insert, replace.

Several item lines may be about the same item. Consequently ADMINS accomodates easily to multiple values (e.g. a person may have more than one occupation) and to relational data (e.g. a record of product contains sub-assembly). The technique for storing multiple values is efficient in that there is no "a priori" reservation of space for all the possible multiple values. Rather the storage structures assimilate all the empirical occurrences. For example, in a record of "person has previous jobs" some people may have none and some may have a dozen. This flexibility also applies to the case where there is missing data or fragmentary data in a file. The identification of items is never achieved via physical positioning but rather through content identifications. Consequently the space occupied by data is proportional only to the amount of actual data present in the file. Also the storage algorithms that write the data values into the pages are designed to perform bit packing and to recognize "runs"-- repeating or ascending sequences of values from item line to item line-- and to store these repetitions as single values plus repetition counts, thereby producing major savings in disk space for sorted data files.

The ADMINS file system is the software for managing data bases of relational records (files). The relational record file norms are used by the software to organize the storage of the data on the disk in the most efficient manner. For example, the normative declaration of the dependencies in a hierarchical classification code is used by the packing routines to recognize runs in the higher order digits of the classification codes, when these codes are applied to cases and stored on the disk. ADMINS as a file system is unique in that its system directories are aware of details of file content as well as the more common file name and location found in conventional file systems. System directories (part of the system log) are managed as part of the data base itself. The file structure used in

ADMINS supports both sequential and non-sequential access methods both via a user written program and via user given command level instructions. The file system is designed on a large scale, i.e. for many files (relational records) and for large individual files. We already have experience of applications with several hundred files, where individual files contain tens of thousands of item lines.

Relational Data Structures

ADMINS is designed to allow the explicit declaration of all categories, codelists and relations that exist in a complex data base, down to the level of a facet within a category and the code within a facet. Each category (field) is named. Groups of named fields may themselves be named. Fields ("facets") in these groups may be in dependent relationship, as in a hierarchecal classification. The contents of a field may be a numerical value, a variable length character string or a nominal code; nominal codes may contain more than one level and may be drawn from multi-level hierarchical classification schemes. The file structure consisting of named categories in relation-- whether used for data or data description or the system log-- is called a relational record.* Also one or more "pictures" assigned to each category in the data base relates the internal binary value of data to an external printable representation. The normative information about the data forms-- the category and relational structure for the data fields in each relational record, which we call the file norms, and the facetted and hierarchecal structure within the categories themselves-- are all contained in the directories of the file system, where they comprise part of the system log. Pictures in ADMINS are similar to pictures in Cobol but contain additional features such as table lookup on (perhaps hierarchical) descriptor lists. Via pictures decimal quantities such as money can be stored and operated upon efficiently as binary bit strings. For example a picture of "$999999V99" may describe a 32 bit binary quantity of dollars and cents.

---

*The term relational record is used to distinguish between a data management approach where the record contains values and the relationship between the values is assumed in the users mind and the ADMINS data management approach where the relationship, as well as the related values, is known to the software. For example, a record may contain a person-id and an occupation code. This code for the description of the particular occupation is usually assumed to be a characteristic of the person, i.e. the relationship is "has" as in the phrase "person has occupation" and remains implicit. But what about the relationships "person desires occupation", "person resigns from occupation", etc. In ADMINS we explicitly declare these relations. Also in ADMINS relations may be more than dyadic, that is the relation may involve more than two "relata". These concepts are discussed in detail in "Computers and Categorization" by the same authors.

Definitional Relationships Managed Within ADMINS

-----Data Description Level-----

Descriptors Organized By Category

Occupation Descriptors
Country Names

Values Organized By Category

Age, Income, Population, Product Sales

Single Level Codes For Descriptors

Code for occupation descriptors
Code for Country names

Facetted categories:

DATE = YEAR MONTH DAY
BLOCK-FACE = NUMBER "thru" NUMBER "of" STREET

Hierarchical multi-level category:

LOCATION = COUNTRY STATE COUNTY CITY
OCCUPATION = MAJOR-CLASS MINOR-CLASS DETAIL
SIC = DIVISION MAJOR-GROUP GROUP INDUSTRY PRODUCT-CLASS PRODUCT

Facetted Hierarchies

Commodity classifications
ZIP codes

-----Actual Data Level-----

Relationship between categories:

PERSON has OCCUPATION
PERSON desires OCCUPATION
SALESMAN receives ORDER from CLIENT for PRODUCT

The relational record data structure is used for data description and the system log as well as the actual object data.

One Data Structure For:

-----Data Description-----

Norm: CODE for OCCUPATION
Specimen: 1.3.2 = attorney
where 1. = professional and 1.3 = legal

Norm: CODE for STATE
Specimen: 1 alabama, 2 arizona, etc

-----Relations Among Data Descriptions-----

"Bill of Materials"
Norm: PRODUCT contains QUANTITY of PART
Specimen: motor-23A 10 phillips-screw-6B

-----Data-----

Norm: PERSON has OCCUPATION
Specimen: jones lawyer, smith doctor

Norm: PRODUCT ordered in QUANTITY by CUSTOMER
Specimen: motor 23A 10 MIT

-----Derived Data and Reports-----

Norm: OCCUPATION has STE/PERSON
where STE means subtotal of existences, that is this is a norm for a derived file produced by aggregation on the record of PERSON has OCCUPATION.
Specimen: professional.medical.doctor 105

Norm: SALESMAN in TERRITORY sold PRODUCT in AMOUNT
Specimen: smith boston motor-23A 340

-----System Data Description-----

Norm: CODE for TYPE
Specimen: 2.2 = nominal hierarchecal

Norm: CODE for ACCESS

Specimen: 1 = read only, 1.1 = display only, 1.2 = aggregate only, etc

-----System Data-----

"File Norm"
Norm: RELATION-SERIAL has CATEGORY
Specimen: 50 salesman
50 territory
50 product
50 amount

The above is a file norm for the first level category structure of relational record 50 which contains data of form "SALESMAN in TERRITORY sells PRODUCT for AMOUNT".

Software and Organizational Implications of One Data Structure

1. ADMINS manages data descriptions including hierarchical classification schemes, usually the province of hard copy references in most data management systems. Also one set of software programs handles both data descriptions and data at both object and system level (economy of implementation) and the one user instruction language manipulates, retrieves and displays both data description and data at both object and system level (economy of use). In many applications where the lists of data descriptions utilize hierarchies and facet schemes, the data description management can be a more difficult and expensive task than the handling of the actual data. Also the intelligent utilization of the dependencies built into the classifications (e.g. commodity classifications) can yield considerable savings in storage.

Much of the design of information systems for substantial applications for a variety of business vocations (e.g. insurance, manufacturing) consists of relating data descriptions, e.g. "bill of materials" relating a product to its component parts, e.g. "allocations to units" in logistics, e.g. a "process layout" in production control, e.g. topics and subject headings in a correspondence file. In ADMINS these use relations are set out as data sets which mediate between the descriptor lists for the data descriptions and the actual data values. Previously the management of this type of planning and control data received little computer assist. In ADMINS these types of files can be flexibly manipulated and used as a management tool.

2. Via the system log stored in relational records, the system software automatically maintains a complete up to the minute data base on the file norms of not only the original data but also of the derived data and of the reports being generated. The same tools used to manipulate, retrieve and display the object records and their descriptions are also useable to manipulate, retrieve and display the system directories, which are quite detailed records of what is going on. This is ADMINS being used to manage itself. This is particularly useful in applications that produce a wide variety of standard and non-standard reports because we have in effect a report management system as well as a "original data" data management system. A typical request would be for a list of all report files which utilized a particular category in their construction, or which dealt with a particular regional breakdown. Also since all reports are stored in relational records in the file system they can be further manipulated after their production without re-deriving the report files each time.

System data is used to set out the various relationships between different files: authorizing document, report file and history file; data sets in time series relation; different history files (e.g. a file organized by product and a file organized by client); different relational records in the same data set (i.e. about the same class of item identity); descriptor records and data records; linkage records and the linked records.

System data can also be used to set out classifications of privacy and security rights of access and to apply these classifications to file norms to control the manipulation and display operations performed by particular users on particular categories in particular records. System data gets down to the detailed level of a category (field) in a file, both for data value and for data descriptor, and to the operations upon the category in the file. We have then a very precise tool for control and authorization.

3. The ADMINS editor provides context editing facilities on relational records, with inputs and edits checked out against both file norms (e.g. category types:nominal, numerical, etc), format control (e.g. Cobol pictures) and data description lists (e.g. permissable occupations). In this way the editor serves as the command level interface to the access methods and to the update operators. (The access methods and the update operators can also be used via subroutine calls in a user written program.) The same editor program can be used to either alter or input data at all levels: description lists, actual data, data about descriptions (e.g. a bill of materials), reports, system data, decision tables, etc. This is an example of the software generality possible within the ADMINS file system.

4. We have implemented several display programs that operate in a general way on relational records. Consequently one package of display programs are used on data stored in relational records at all levels in the system. That is to print a system directory, a data description list or actual data, before or after aggregation, the same program is used. An example of a display program is the PRTABL program which formats a relational record into an analytic report, preparing a multi-dimensional table (e.g. a sales analysis report) where the user decides, when he gives the PRTABL command, the assignment of control categories to row, column or to the sub-tables within each cell and where data desciption lists and the "pictures" maintained in the system directory provide the labels and data formats automatically.

Command Level Language

The command level language is concerned with re-organization, selection and aggregation of item lines in individual relational records and with relations logic operations working on more than one relational record at a time. (An individual relational record can be an entire file. For example, a telephone and street address directory is a relational record of form "PERSON lives at ADDRESS and has TELEPHONE NUMBER". Or a relational record can simply be a report or a data description list.) By appropriate combination of the various selection and aggregation options, with the facility to re-order and/or remove categories at each step ("rotation", "contraction"), and with relational operations for linking different files via common categories, the non-programmer user can produce a wide variety of derived files and reports in a stepwise interactive mode, without writing any ad hoc programs and with continuous feedback via printout of (samples of) the output at each step. The result of any command level operation may itself be produced as a relational record. Therefore all reports in ADMINS are not only hard copy printouts but can also be available for further manipulation as relational records. Also the user has access in his command level instruction syntax to all the "structure" contained within a category. For example, if a relational record contains DATE in its norm the user can refer to YEAR, MONTH and DAY or if LOCATION is in the norm the user can refer to COUNTRY, STATE, COUNTY and CITY in his command level instructions or if SIC (Standard Industrial Classification) is in the norm the user can refer to DIVISION, MAJOR-GROUP, INDUSTRY, PRODUCT-CLASS, ETC.

Command Level Operations*

OPERATE	on a relational record to produce a new record or a printout ... to select (index) item lines by matching, inclusion, substring, inequality ... to aggregate existences or values under one or several controls over one or several categories ... to select maxima or minima under one or several controls over one or several categories ... to control the choice, placement, and order of categories to be maintained in the output ("rotation", "contraction")
EDIT	a relational record ... to input fresh data into a new relational record ... to append fresh data to an existing relational record ... to find, change, replace, insert a value or an entire item line without file recopying
PRINT	applied to a relational record ... selectively on data descriptions for the record ... selectively on data in the record ... the entire relational record ... via selections by the "operate" interface ... in a multi-dimensional analytic format

---

*Each of these operations are useable not only on the object data, including reports-- but also on data description files and on the system log.

Relations Logic on two or more relational records

... comparing on one or more common categories

... using "and" logic for matching operations within a data set (binding) and cross reference operations between data sets (linking)

... using "or" logic for merging and permuting operations

... using "exclusive or" logic for negation and audit operations

... using "and" logic plus category removal for inference operations

COMPUTE a value for each item line

... to process an arithmetic formula on the values in the item lines

... to compute a new category for the data base

RECODE a relational record

... to process a decision table recode structure against item lines in a relational reord

Let us illustrate the command level operations with an example. In ADMINS relational records can be kept in sort on the data values. One command level operation is concerned with re-ordering the categories (fields) in the norm of a relational record and sorting the values in the item lines based on the new category order. This operation of re-organizing a file by re-ordering the categories in it can be applied at all levels of information management.

1. Object data. Re-ordering the categories in a

---

telephone directory might be used to produce a derived directory sorted by telephone number. Re-ordering the categories in a file of invoices ordered by invoice number might be used to produce a report file ordered by product.

2. Data Description. Re-ordering the categories in a codelist of occupation (i.e. a record of "OCCUPATION has DESCRIPTION" with a specimen item line of 1.3.2 = accountant) would be used to produce an alphabetized code list. Re-ordering the categories in a bill of materials record might be used to produce the first level of a parts where used report.

3. Reports. Re-ordering the record of a cross-tabulation of salesman .vs. territory .vs. product (i.e. a norm of "SALESMAN in TERRITORY sold PRODUCT in AMOUNT"-- itself a record derived via aggregation of sales orders) could yield a report whose lines were organized (sorted) by product or territory or amount (the cell value itself) as opposed to salesman.

4. System Data. Re-ordering the system data record that contains the first level category structure for all relational records in the data base (a norm of "RELATION-SERIAL has CATEGORY") produces a system level report record ordered by the category (field) name for each object level relational record in the data base-- a very useful tool for finding derived data or reports.

Programming Languages

A programming language facility is required to meet need for complex reports and intricate updating that a general purpose command language facility cannot handle; also to more efficiently package procedures developed at command level using stepwise methods, and for model building and simulation. Also a programming language facility can be used to build customized command level enhancements for particular applications.

The data management system itself is written in an extended algol-like language developed by the ADMINS Project, which we call ADMINS Language (AL). Adding commands to the system consists of writing programs in this language which conform to the conventions set up within the data management system programs. However, algol-like languages are complex languages and AL in particular with its extensions would be difficult for non-senior programmers to use. Therefore we have implemented a COBOL interface to our data management system which we call the COBOL translator. The COBOL translator produces executable tables from Cobol source programs after checking out data usages in the data division of the Cobol source against the resident system directories. The Cobol translator also updates new relation norms into the system directories and can translate high level update verbs used in the Cobol for accessing ADMINS data structures into appropriate file system calls.

The Cobol translator simplifies the programmers task in operating with a particular program on the resident data base and in adding a derived file to the data base. The programmer declares the norms for the files he is accessing or creating in the data division of the Cobol source program and the Cobol translator relates these declarations to the resident system directory. Also the Cobol runtime package and the ADMINS system software automatically handle the actual I/o and the clerical details of data arrangement in core, when the Cobol program is run. The COBOL executor, the program that runs the compiled executable tables on actual data, contains an interactive debugging facility which includes symbolic breakpoint setting and on-line display of data via symbolic name with picture interpretation.

Size of Data Base

Our initial system stores about 10 million bytes of data on its disk pack. Different data bases are kept on other disk packs or on tape. Additional disk drives may be added to the system to enlarge the on-line data capacity through built in hardware capability and existing software provisions.

ADMINS is fundamentally data management software that is implemented currently on a mini-computer. Past implementations of ADMINS have been on maxicomputers; future maxicomputer versions are planned. At no time have we limited our designs to small data bases or a small computer. The size and complexity of a data base that ADMINS could manage is a function of the power of the computer host and available disk. In the case of the SEL 7200 several 2314 (30 million byte capacity) type disks seem plausible.

The maximum size of a paged segment of a relational record in current ADMINS is about a million bytes. However a relational record norm may control several paged segments. Also ADMINS storage structures utilize bit packing and compress runs of values to an initial value plus a repetition count. Consequently a million byte segment in ADMINS is probably the equivalent of several million byte segments in another system.

As ADMINS uses relational records (files) to store descriptor lists, reports, system log, etc, as well as "master files", the system necessarily is able to accomodate several thousand relational records in its current implementation, many of which in any given application will be of modest size.

ADMINS Installation

Our host computer is the SEL 7200 minicomputer. We chose the 7200 for its larger address space (128k bytes), hardware virtual memory, suitability for time sharing*, and the efficiency of its instruction set in the use of memory.** We were well aware when we purchased the 7200 that the traditional minicomputer software provided by the vendor-- assembler, loader, Fortran, Basic, device management, etc-- would not be sufficient to host software of the complexity we were contemplating. In order to successfully manage the implementation of ADMINS, especially within a minicomputer framework, a powerful programming management tool was needed. We developed ADMINS Language (AL) and the ADMINS Operating System (AOS) to meet this need. (A control program to offer general purpose time shared access to these facilities is under developement.) AL is an algol-like language which under AOS supports character string handling, dynamic storage allocation, dynamic loading and linking of subroutines, and powerful systems programming tools such as bit and byte manipulation, access to data addresses and actual use of assembly language in the procedure language program itself. AOS provides device management, a file system with both sequential and random access files, a powerful context editor for preparation of source programs and test data bases within the AOS file system, a documentation preparation package (on which this paper was prepared), and powerful debugging features including symbolic inspection of programs at the compiler level during execution.

We also have an index sequential package and a sort package so that AL, AOS and, if required, the Cobol interpreter (lifted out of the data management system), can provide an environment for particular turnkey applications without involving the rest of the ADMINS data management system, or can be used in those situations where a more traditional approach is preferred as an initial stage into data base management on a minicomputer. Although a secondary purpose in our research, we nevertheless consider the development of an applications building environment-- extended algol-like systems language, Cobol, file handling including index sequential and sort, many utilities, flexible execution-- on inexpensive hardware both a significant and highly practical tool for the state of the art in developing inexpensive systems.

---

*7200 features include hardware memory protect and fragmentation of programs down to the page level, hardware paging, and supervisor, master and slave modes of operation.

**Notably the ability to reference the CPU registers as the first eight memory locations, and the ability to instruct absolute, self-relative, and pre- and post-indexing in a single 16 bit instruction word.

Conclusion

A general purpose data handling method operating on modestly priced hardware has been described. The use of such a facility seems appropriate to such diverse areas as production control, marketing analysis, manpower as well as vocational applications such as hospital, municipal and data oriented research and in general where methods for routine processing solely of actual data values maintained in large monolithic files, where description lists, reports and relational data are managed by people based hard copy methods, is proving inadequate. The developement of data management software** designed primarily for use by non-programmer systems and methods people that operates on modestly priced mini-computer hardware opens up many choices over the next few years for the data oriented computer user.

**Both the logical specifications that could be applied to any computer of modest size on up, and the actual programs for a particular computer.

ADMINISTRATIVE APPLICATIONS OF APL

Moderator: Adin Falkoff
Manager
IBM Scientific Center
Philadelphia, Pennsylvania

CAUSE MEETING, Dec. 11, 1973
SESSION ON
ADMINISTRATIVE APPLICATIONS OF APL

Moderated by
Adin Falkoff
Manager, IBM Scientific Center
Philadelphia, Pa.

ABSTRACT

The use of APL for academic purposes is well known, and has been a key factor in the spreading use of this mathematically-based programming language. It is less well known, however, that the characteristic generality and conciseness of APL make is an important language for business data processing. In these sessions, speakers will discuss the use of APL for terminal-based on-line school administrative applications, noting, in particular, how easy programming influences such applications. The issues to be discussed include the relationship between interactive and batch processing of the same data, use of bulk output devices, user languages for non-programmers, and security questions arising from the use of terminals for accessing sensitive information.

This document was edited and composed with an APL text handling package, and sent to a remote line printer from an APL terminal.

ADMINISTRATIVE APPLICATIONS OF APL

Moderator: Adin Falkoff
Manager
IBM Scientific Center
Philadelphia, Pa.

First Hour:

"Administrative Uses of APL at Syracuse University"
Phyllis Kent
Assistant Director for Academic Computing Services
Syracuse University
Syracuse, New York

"Non-Numeric Use of APL in the University Environment"
L. P. A. Robichaud
Director of Computer Center
Universite Laval
Quebec City, Quebec

"Using APL to Provide a Computer Utility for Education"
Becky Willis
President
SECOS - Shared Educational Computer System
Poughkeepsie, New York

Second Hour:

"An On-Line MIS to Assist the Management of Facilities"
S. C. Fallis
D. D. McElroy
E. E. Payne
Planning and Management Systems Division
University of Texas at Dallas
Dallas, Texas

"An APL Solution to a Data Processing Problem"
Robert E. Schaulis
Director of Information Services
Coast Community College District
Costa Mesa, California

"Interactive Analysis of Student Records"
P. C. Berry
Manager of APL Education
IBM Scientific Center
Philadelphia, Pa.
R. M. McCarthy
Research and Technical Services
Computing Center
State University of New York
Binghamton, New York

Third Hour:

"Security Facilities in an APL System"
Richard Cox
APL Services, Inc.
Houston, Texas

Round Table Discussion: Questions and Answers on Interactive Administrative Applications

ADMINISTRATIVE USES OF APL AT SYRACUSE UNIVERSITY

Phyllis Kent
Assistant Director
Academic Computing Services
Syracuse University

ADMINISTRATIVE USES OF APL AT SYRACUSE UNIVERSITY

Phyllis Kent
Assistant Director for Academic Computing Services
Syracuse University
Syracuse, New York

In the past five years a number of APL applications have been proposed as solutions to problems that arise in administering the affairs of the University. Many of the proposals have been implemented, and exist now as working systems that are used regularly to assist administrators in their work. Prior to availability of external files to our APL system, projects were confined to small scale departmental or univeristy office tasks which had small data bases, but access to external files from APL in 1971 made several university-wide projects with large data bases a reality at Syracuse.

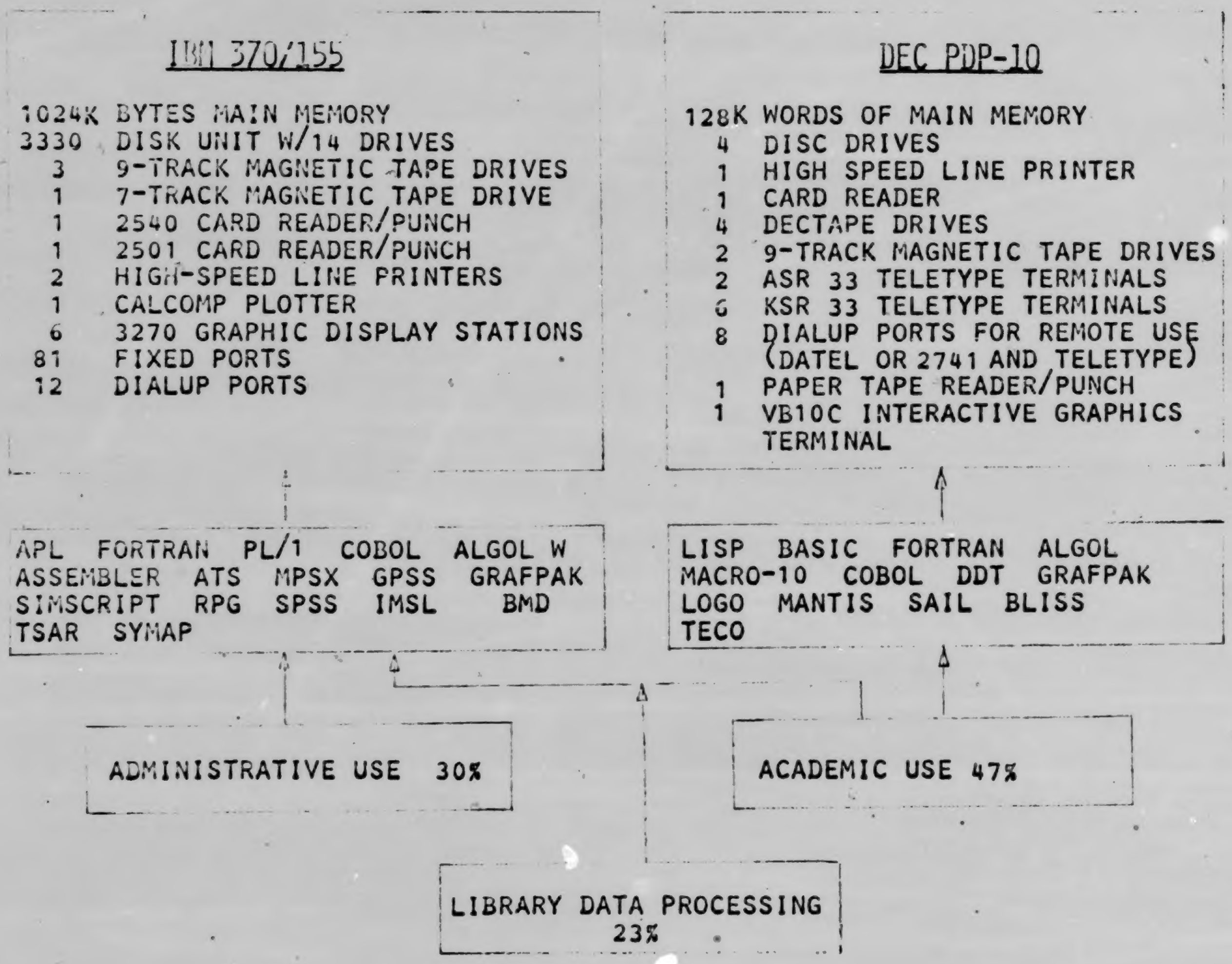
The community of people who are served by these systems consists of 10,000 undergraduate students, 5000 graduate students, 900 full-time faculty and 400 full-time professional and administrative staff. The computer resources available at the academic computing facility on the main campus and the principal features of our APL system are shown in Figures 1 and 2.

Some of the administrative systems which are written in part or entirely in APL are listed in Figure 3. These systems serve the entire university community in some sense. They have in common the use of APL for interactive input or updating of information and for interactive inquiry from university offices. The first five systems in this list are working systems. Some combine APL interactive processing with batch processing of the same data. The last three systems have been designed but not yet implemented.

The Faculty Profile System is representative of the systems in use at Syracuse. It combines the best features of APL for ease of programming and interactive access to data with the batch facilities of our computer. Figures 4 through 15 describe in detail the features of this system. The objective of the Faculty Profile is to provide on one file current data on faculty, to produce regular annual reports on faculty tenure, salaries, age, years of service, and academic activity. A flexible and easily modified inquiry system provides the Office of Academic Affairs with ready interactive access to information on faculty. Figures 8, 9 and 10 contain a sample of an interactive file update session as it might be executed in the Personnel office of the university. Figures 12, 13 and 14 show a typical interactive inquiry session as it might be executed in the Office of Academic Affairs. Figure 16 describes the features of the APL system at Syracuse that protect the security of sensitive data and the integrity of the APL programs that access those data.

In summary Figure 17 outlines the features of APL that make it most suitable for administrative applications.

SYRACUSE UNIVERSITY COMPUTING CENTER
ACADEMIC COMPUTING FACILITY



IBM 370/155

1024K	BYTES MAIN MEMORY
3330	DISK UNIT W/14 DRIVES
3	9-TRACK MAGNETIC TAPE DRIVES
1	7-TRACK MAGNETIC TAPE DRIVE
1	2540 CARD READER/PUNCH
1	2501 CARD READER/PUNCH
2	HIGH-SPEED LINE PRINTERS
1	CALCOMP PLOTTER
6	3270 GRAPHIC DISPLAY STATIONS
81	FIXED PORTS
12	DIALUP PORTS

DEC PDP-10

128K	WORDS OF MAIN MEMORY
4	DISC DRIVES
1	HIGH SPEED LINE PRINTER
1	CARD READER
4	DECTAPE DRIVES
2	9-TRACK MAGNETIC TAPE DRIVES
2	ASR 33 TELETYPE TERMINALS
6	KSR 33 TELETYPE TERMINALS
8	DIALUP PORTS FOR REMOTE USE (DATEL OR 2741 AND TELETYPE)
1	PAPER TAPE READER/PUNCH
1	VB10C INTERACTIVE GRAPHICS TERMINAL

APL FORTRAN PL/1 COBOL ALGOL W ASSEMBLER ATS MPSX GPSS GRAFPAK SIMSCRIPT RPG SPSS IMSL BMD TSAR SYMAP

LISP BASIC FORTRAN ALGOL MACRO-10 COBOL DDT GRAFPAK LOGO MANTIS SAIL BLISS TECO

ADMINISTRATIVE USE 30%

ACADEMIC USE 47%

LIBRARY DATA PROCESSING 23%

APPROXIMATELY 115 IBM 2741, DATEL AND OTHER SLOW SPEED REMOTE TERMINALS ON CAMPUS.

FIGURE 1

SYRACUSE UNIVERSITY
APL SYSTEM

5734-XM6 VERSION OF APL - IBM
APLPLUS LEVEL 1 WITH FILESUBSYSTEM - STSC

SPECIAL S.U. FEATURES:

APL IJR - INTERNAL JOB READER FOR SUBMITTING BATCH JOBS FROM REMOTE TERMINALS

APLIN - UTILITY TO CREATE APLPLUS FILES FROM OS DATA SETS

APLOUT - UTILITY TO CREATE OS DATA SETS FROM APLPLUS FILES

OPERATING SYSTEM: SUOS
SYRACUSE UNIVERSITY OPERATING SYSTEM
A MODIFIED OS/MFT

FIGURE 2

ADMINISTRATIVE APPLICATIONS
AT
SYRACUSE UNIVERSITY

SYRACUSE UNIVERSITY LIBRARY DATA MANAGEMENT SYSTEM
FACULTY PROFILE SYSTEM
SYSTEM MAINTENANCE MONITOR
SYRACUSE UNIVERSITY APL POSTAL SYSTEM
GRADUATE SCHOOL ENROLLMENT
FILM LIBRARY SCHEDULING
OFFICE OF RESIDENTIAL LIFE ROOM AND BOARD PROJECT
ART COLLECTION INVENTORY

FIGURE 3

FACULTY PROFILES

FACULTY EMPLOYMENT AND ACADEMIC STATUS SYSTEMS (FEASS)

A PROJECT OF THE OFFICE OF INSTITUTIONAL RESEARCH
WITH ASSISTANCE OF PERSONNEL DEPARTMENT
AND OFFICE OF VICE CHANCELLOR FOR ACADEMIC AFFAIRS

RON KURDZIEL
MARY PAT OLIKER
JOSEPH WENZEL

FIGURE 4

OFFICE ACCESS TO INFORMATION

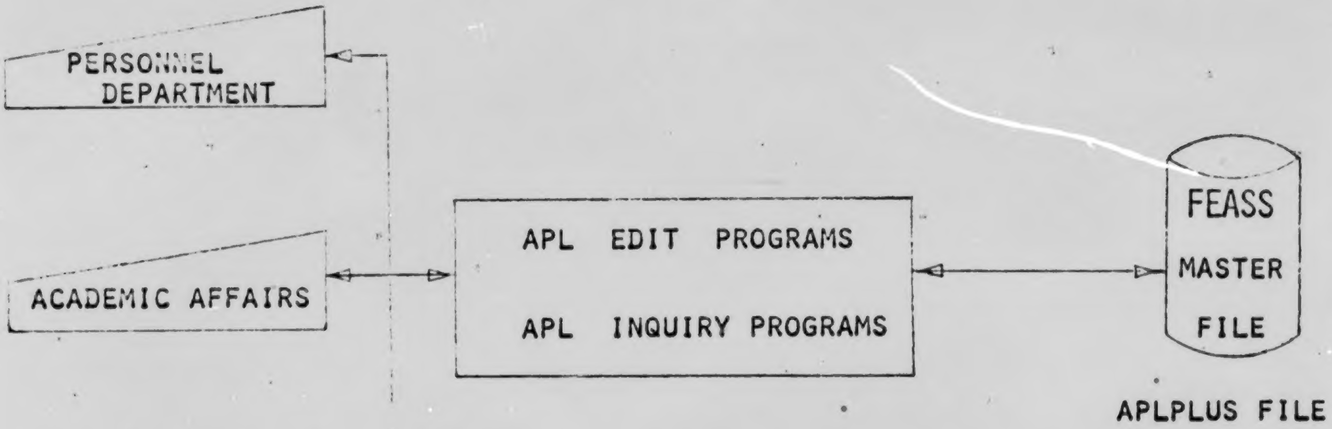

PERSONNEL DEPARTMENT
ACADEMIC AFFAIRS
OTHER
APL EDIT PROGRAMS
APL INQUIRY PROGRAMS
FEASS MASTER FILE
APLPLUS FILE

FIGURE 5

CONTENT OF FEASS MASTER FILE

APLPLUS FILE:	1000 RECORDS OF 480 CHARACTERS EACH COMPONENTS ARE 15 BY 480 CHARACTER MATRICES EACH RECORD CONTAINS INFORMATION ABOUT A SINGLE FACULTY MEMBER RECORDS IN SOCIAL SECURITY NUMBER ORDER
GENERAL INFORMATION:	SOCIAL SECURITY NUMBER, NAME, BIRTH DATE, SEX, MARITAL STATUS, MINORITY STATUS
HIGHEST DEGREE INFORMATION:	AWARDING INSTITUTION, DATE AWARDED, MAJOR FIELD
PRESENT STATUS INFORMATION:	RANK, FULL/PART TIME, CONTRACT TYPE, BASE SALARY, DATE OF APPT. TO PRESENT RANK, DEPARTMENT CODE, ETC.
PREVIOUS FISCAL YEAR INFORMATION:	RANK, BASE SALARY, ETC.
STARTING APPOINTMENT INFORMATION:	
TENURE STATUS	
LEAVE STATUS	
BENEFITS:	TIAA, HEALTH INSURANCE, LIFE INSURANCE, TUITION DEPENDENT SCHOLARSHIP INFORMATION, ETC.

FIGURE 6

COLLECTION OF INFORMATION

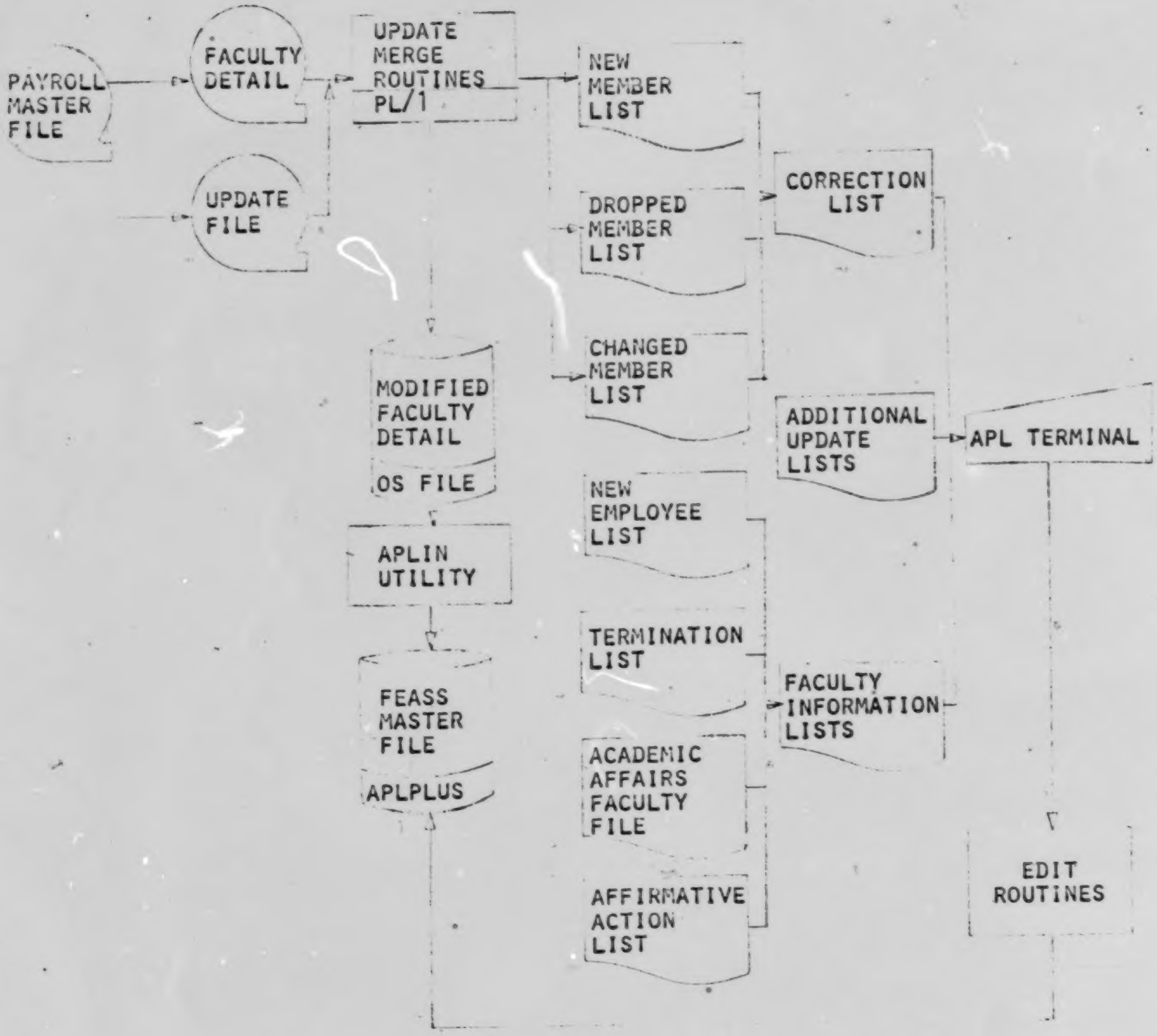

PAYROLL MASTER FILE
FACULTY DETAIL
UPDATE FILE
UPDATE MERGE ROUTINES PL/1
NEW MEMBER LIST
DROPPED MEMBER LIST
CHANGED MEMBER LIST
NEW EMPLOYEE LIST
TERMINATION LIST
ACADEMIC AFFAIRS FACULTY FILE
AFFIRMATIVE ACTION LIST
CORRECTION LIST
ADDITIONAL UPDATE LISTS
FACULTY INFORMATION LISTS
APL TERMINAL
EDIT ROUTINES
MODIFIED FACULTY DETAIL
OS FILE
APLIN UTILITY
FEASS MASTER FILE
APLPLUS

FIGURE 7

```
ENTER TAGS

→"NAM;RNK;BAS;PAY"
FIELDS:
NAM   RNK   BAS   PAY


INPUT SOC.SEC.NO.

→S000000008
RECORD: HYDE,CLYDE                        000000008

→[?


NAM       HYDE,CLYDE
RNK       110300
BAS       15000
PAY       17
→[HYDE,CLYDE,L;120000;12000;17]

INPUT SOC.SEC.NO.

→S000000008
RECORD: HYDE,CLYDE,L                      000000008

→[?


NAM       HYDE,CLYDE,L
RNK       120000
BAS       12000
PAY       17
→["

ENTER TAGS

→"RNK;DC1"
FIELDS:
RNK   DC1


INPUT SOC.SEC.NO.

→S000000017

SOC. SEC. NO. 000000017 INVALID
```

FIGURE 8

```
→S000000012
RECORD: MEYERS,CAROL                   000000012

→[12345;1326]
***RNK INVALID***
VALUE ENTERED: 12345

REENTER RNK
→120000
***DC1 INVALID***
VALUE ENTERED: 1326

REENTER DC1
→987
***DC1 INVALID***
VALUE ENTERED: 987

REENTER DC1
→⌶
RECORD: MEYERS,CAROL                   000000012

→[]

INPUT SOC.SEC.NO.

→S"

ENTER TAGS

→"NAME;SEX;RANK"

IMPROPER DATA FORMAT
LINE CANCELLED

ENTER TAGS

→""

PROCESSING TERMINATED
```

FIGURE 9

```
ENTER TAGS

→"RNK;SEX;NAM;XXX"
INVALID TAGS
XXX

ENTER CORRECT TAG OR A CARRIAGE RETURN
XXX:    DC1
FIELDS:
RNK   SEX   NAM   DC1


INPUT SOC.SEC.NO.

→$"

ENTER TAGS

→"RIK;SEX;NAM;XXX"
INVALID TAGS
RIK
XXX

ENTER CORRECT TAG OR A CARRIAGE RETURN
RIK:    RNK
XXX:
FIELDS:
RNK   SEX   NAM


INPUT SOC.SEC.NO.

→$"

ENTER TAGS

→""

PROCESSING TERMINATED
```

FIGURE 10

DETAIL OF INQUIRY AND REPORT SYSTEM

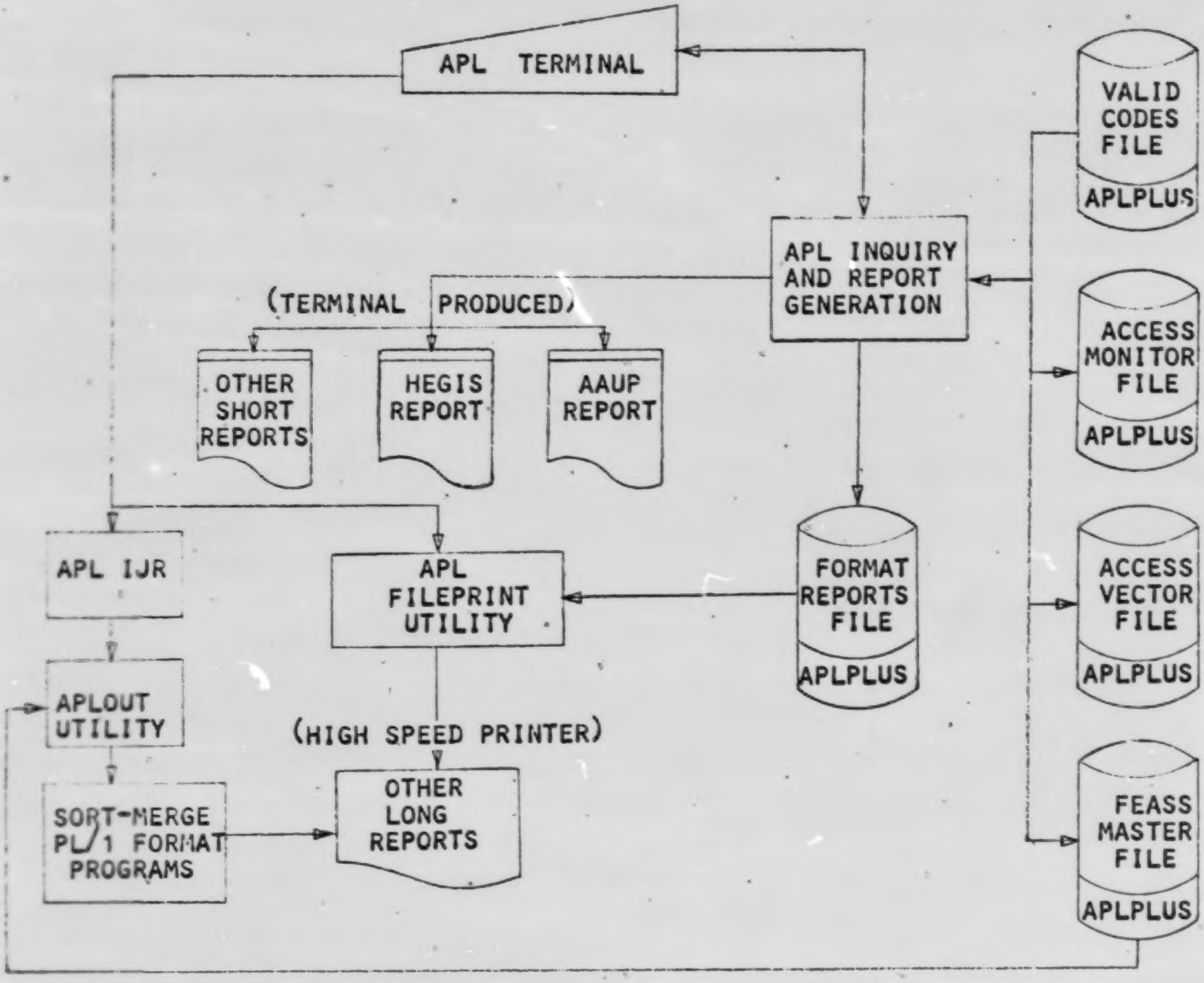

APL TERMINAL
VALID CODES FILE
APLPLUS
APL INQUIRY AND REPORT GENERATION
(TERMINAL PRODUCED)
ACCESS MONITOR FILE
APLPLUS
OTHER SHORT REPORTS
HEGIS REPORT
AAUP REPORT
APL IJR
APL FILEPRINT UTILITY
FORMAT REPORTS FILE
APLPLUS
ACCESS VECTOR FILE
APLPLUS
APLOUT UTILITY
(HIGH SPEED PRINTER)
SORT-MERGE PL/1 FORMAT PROGRAMS
OTHER LONG REPORTS
FEASS MASTER FILE
APLPLUS

FIGURE 11

SAMPLE INTERACTIVE INQUIRY SESSION

```
      HOW MANY UNMARRIED FACULTY WOMEN MAKE OVER $10,000 A YEAR?  ⍝

      COUNT (SEX EQ 'F') AND (MARRIAGE EQ 'S') AND PAY GT 10000
2


      WHICH FACULTY WOMEN MAKE OVER $10,000 A YEAR?   ⍝

      PRINT (SEX EQ 'F') AND PAY GT 10000
008223709 LASEN, PATRICIA
010182058 RAPHAEL, CYNTHIA
015055184 MEYER, JANE K.
022242877 MACY, JEAN
025367410 ENDERS, RUTH L.
027240033 DELANEY, JEANETTE
029385943 JOHNSON, BETTE
031129862 LESNIK, MARGARET
041502490 ADAMS, MARY ANN
044246387 ROBERTS, HELEN


      COMPARE SALARIES OF WOMEN AND MEN   ⍝

      COUNT (SEX EQ 'F') AND PAY GT 10000
10
      COUNT SEX EQ 'F'
14
      COUNT (SEX EQ 'M') AND PAY GT 10000
72
      COUNT SEX EQ 'M'
76

      10÷14
0.7142857143
      RATIO FOR WOMEN   ⍝

      72÷76
0.9473684211
      RATIO FOR MEN   ⍝
```

FIGURE 12

```
      SUM (SEX EQ 'F') SELECT PAY
153615

      153615÷14
10972.5
      AVERAGE PAY FOR WOMEN   ⍝

      SUM (SEX EQ 'M') SELECT PAY
1349333

      1349333÷76
17754.38158
      AVERAGE PAY FOR MEN   ⍝


      COMPARE FACULTY AGES   ⍝

      MAX AGE
61

      COUNT (SEX EQ 'F') AND AGE GT 50
3
      COUNT (SEX EQ 'M') AND AGE GT 50
19

      SUM (SEX EQ 'F') SELECT AGE
521

      521÷14
37.21428571
      AVERAGE AGE FOR WOMEN   ⍝

      SUM (SEX EQ 'M') SELECT AGE
3232

      3232÷76
42.52631579
      AVERAGE AGE FOR MEN   ⍝
```

FIGURE 13

```
        WHAT IS THE HIGHEST SALARY FOR FACULTY WOMEN?   ∩

        MAX (SEX EQ. 'F') SELECT PAY
21900

        WHO IS SHE?   ∩

        PRINT (SEX EQ 'F') AND PAY EQ 21900
031129862 LESNIK, MARGARET

        HOW MANY FACULTY MAKE WITHIN 100 OF THAT?   ∩

        COUNT (PAY GT 21800) AND PAY LT 22000
1

        WHO IS NEXT LOWER IN PAY?   ∩

        PRINT PAY EQ MAX (PAY LT 21900) SELECT PAY
002186962 BANNON, PAUL


        WHO IS HIGHEST PAID FACULTY MEMBER?   ∩

        PRINT PAY EQ MAX PAY
027264313 ROGERS, ALLEN K.

        MAX PAY
36400
```

FIGURE 14

BACKUP AND ARCHIVE FACILITIES

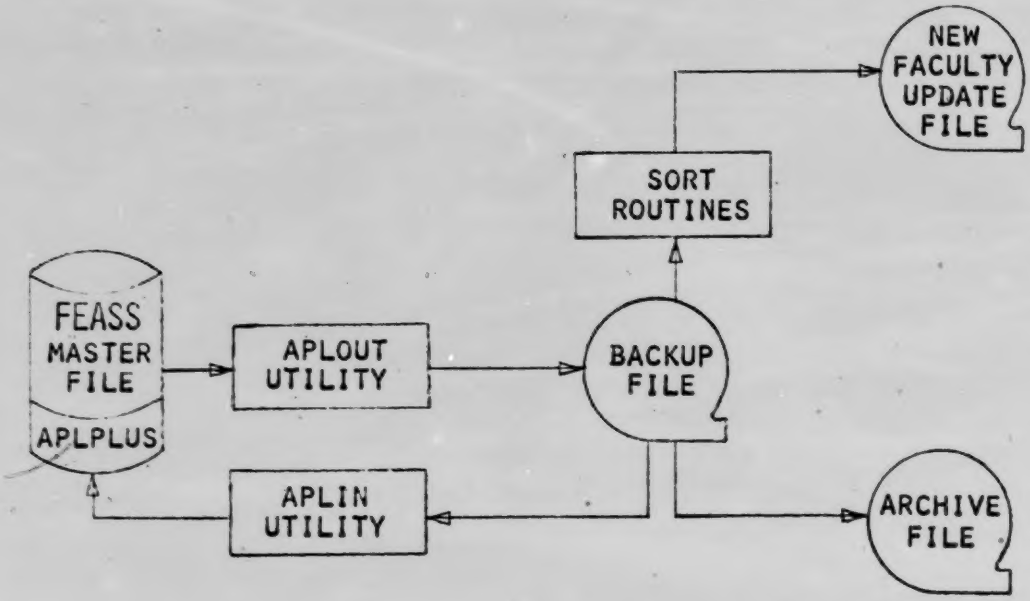

NEW FACULTY UPDATE FILE
SORT ROUTINES
FEASS MASTER FILE
APLPLUS
APLOUT UTILITY
BACKUP FILE
APLIN UTILITY
ARCHIVE FILE

FIGURE 15

SECURITY OF SENSITIVE DATA

SYRACUSE UNIVERSITY APLPLUS SECURITY FEATURES

APLPLUS FILE PROTECTION:

FILE IS PROPERTY OF ACCOUNT THAT CREATED IT
CREATOR CONTROLS ACCESS
FILE CAN BE SHARED WITH ANY OTHER ACCOUNTS
ANY TYPE OF ACCESS CAN BE AUTHORIZED
PASSNUMBER CAN BE ATTACHED TO AUTHORIZED ACCOUNT

PROGRAMMABLE SECURITY MEASURES:

ACCESS ONLY TO VALID SET OF ACCOUNTS
ACCESS ONLY TO CERTAIN FILE ITEMS
ACCESS ONLY FROM CERTAIN TERMINALS OR CLASSES OF TERMINALS
ACCESS ONLY ON CERTAIN DAYS, TIMES OF DAY
LOG EVERY ACCESS IN A MONITOR FILE

PROTECTION OF PROGRAM PACKAGES:

SEAL USER COPY OF WORKSPACE TO LOCK ALL FUNCTIONS
 AND PREVENT COPY, DISPLAY OR ERASURE OF ANY PART
USE WALKAWAY FEATURE TO FORCE SIGN-OFF IN CASE OF
 PROGRAM INTERRUPT. PREVENTS DISPLAY OR MODIFICATION
 OF CONTENTS OF WORKSPACE

434

FIGURE 16

WHY APL?

1. NEED FOR INTERACTIVE FACILITY TO UPDATE AND QUERY FILES AT SEVERAL UNIVERSITY OFFICES.
2. PROGRAMS MUST BE CONVERSATIONAL AND/OR EASY TO USE BY ADMINISTRATIVE PERSONNEL.
3. NUMBER AND TYPE OF REPORTS AND QUESTIONS CAN NOT BE PREDICTED IN ADVANCE. IT MUST BE POSSIBLE TO CREATE PROGRAMS FOR THESE EASILY ON SHORT NOTICE.
4. METHODS FOR GIVING RESTRICTED ACCESS TO SENSITIVE DATA AND FOR PROTECTING INTEGRITY OF FILES MUST BE READILY AVAILABLE.

FIGURE 17

NON-NUMERIC USE OF APL IN THE UNIVERSITY ENVIRONMENT

L. P. A. Robichaud
Director of Computer Center
Universite Laval
Quebec City, Quebec

NON-NUMERIC USE OF APL IN THE UNIVERSITY ENVIRONMENT

L. P. A. Robichaud
Director of Computer Center
Universite Laval
Quebec City, Quebec

ABSTRACT

The presentation will touch briefly upon a number of applications which are not primarily numeric in nature. Some of these include the use of a general text-editor for inputting, retrieving, correcting and outputting relatively large text databases in arbitrary character representation (containing any combination of overstruck characters). One project consists of a bank of questions for preparing medical exams (having a specified distribution of questions) and the correction of these exams. Another project deals with an APL-based bank of all course and program offerings in the University. Finally, another project deals with a mini-MIS system developed by a school of administration.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

USING APL IN AN EDUCATIONAL COMPUTER UTILITY

Rebecca E. Willis
President
Shared Educational Computer System (SECOS)
Poughkeepsie, New York

Using APL In An Educational Computer Utility

by

Rebecca E. Willis
Shared Educational Computer System, Inc.
Poughkeepsie, New York

The Shared Educational Computer System is a not-for-profit computer utility for educational purposes. It is presently serving twenty-eight different educational institutions. These institutions represent a full academic range, from grade school through graduate school, including medical schools, graduate schools, four-year colleges, two-year colleges, community colleges, research organizations, and other non-profit organizations.

APL and Coursewriter III are available on the system 23 hours a day, 7 days a week. Schools pay a fixed monthly fee for unlimited use of the system. Large public libraries of programs are maintained on the system for free access by all users.

Over a period of two years we have grown from an instructional system to a general purpose Educational utility using APL. Initially the services were used by most of the schools primarily for instructional purposes. However, schools that have been on the system for several years have a growing body of administrative applications, both in use and under development. This body of administrative programs is growing at an accelerating rate. The pricing structure has encouraged schools to develop administrative applications to offset the cost of the instructional usage. One of the largest installations is completely paid for by administrative use in the day time, and the same terminals are made available to students in the evening at no additional cost.

The administrative programs on the system cover a wide variety of topics and applications. We use the system quite extensively in our own office. We have an interactive accounting system which does standard double entry accounting for all of our bookkeeping work. This includes general ledger, journals, cash disbursements. There are two auxiliary packages which feed information into the large and more general accounting package. The first is a billing program which produces all of the bills for use of the system, and the second is a payroll program which computes the payroll, produces the checks and posts the necessary entries into the accounting system. Within the office we make extensive use of text editing in APL, producing a number of newsletters and brochures, as well as publishing textbook materials relating to instructional use of computers.

Our Electronic Post Office provides flexible and convenient communication between ourselves and all of the locations, as well as a communication link between the users of like interests on the system. In addition to the ability to place messages for later retrieval from a common file, each user is notified at sign on time if he has a message pending in his post box.

The many administrative uses that are made of the system in remote locations are difficult for us to keep track of since all work is initiated from the individual school's terminals. Frequently things are going on that we at the central office are not aware of. This was brought sharply to our attention in a telephone call from one of the schools using the system. It had been our impression that this school was using the system only for instructional use, but the call was a desperate one from a business manager saying that although someone was on the way to repair their terminal they weren't sure that they could wait the two hours. It was payroll day, and they were anxious to be able to run their payroll on the terminal.

We have one school board which has been using the system for teacher negoiations. A package of programs prepared by one of the school board members allows the school to compute the total financial impact on the school of each item as it is presented as a possibility at the meeting. The presence of the terminal within the room as teacher negoiations are carried out has totally changed the nature of the discussions from questions about the validity of the figures to making the decision about the appropriate action for the school and for the teachers, with very little time being spent on discussing the validity of the figures.

Many of the programs which are in use are financial in nature and cover areas that are particularly interactive or where reports on demand is particularly important. These include such items as budget preparation, records of grants and contracts, records and reports of deans' special funds. Applications such as the catalog of faculty and letters of appointment is usually kept on line only at the beginning of each semister and then archived until the following semister so that it can be updated and correct hard copy produced for the appropriate offices.

Three types of inventory programs are in use*. One is an inventory of capital equipment which includes all information necessary for reporting on the condition and whereabouts of the equipment that was attained through special funding. The second is a space inventory which allows scheduling of special space facilities, as well as a total inventory of space available.

Presently under development is a general stores inventory for a medical school, which includes billing capability, as well as ordering capability. A program for generating patient discharge summaries from the Obstetrics Department includes the generation of english language reports from checklist summaries which are prepared by the discharging doctor. In addition statistical analysis of the characteristics of all summaries from that department. This program is in the process of being generalized for use in several other departments. Also under development is putting on-line the clinical chemistry lab facilities of the medical school. This involves taking paper tape output that is produced from the chem lab equipment and reading it into the APL system from a terminal with a paper tape reader.

At another medical installation on our system a cancer research project is going forward. In this case the system will be used for management of the research project as well as for analysis. The APL system will trace ten thousand patients for a period of five (5) years, notifying members of the study of visits they must make to the center for tests and compiling test results and statistical analysis of the results of the study.

A financial aid package is used at several of the institutions which produces on demand a summary of all the students within the institution receiving financial aid according to any of the available parameters that the financial aid office wish to choose.

Also under development is an on-line interaction with a file containing information of all of the employment locations within New York State. This will be used both for placement information for colleges and for curriculum decisions within technical curriculum. A historical study which was developed for research purposes is being made available to the schools on the system and as a very diverse point from the many quite technologically oriented projects which are on-going on the system.

With this wide variety of administrative programs, one must ask the question, "Why do people write these types of administrative programs in APL?" The answers appear to fall into three categories. First, that it was the language available and that the fixed monthly charge on the SECOS system as opposed to an hourly charge has encouraged administrative use of the system. Clearly, the availability of a language must be an important part of any decision as to what language should be used. The second group of answers relate to the interactive nature of APL and the advantages that this gives in administrative applications. Terminals are frequently located in the offices of financial officers, deans, registars: the offices of the people that are responsible for the information. When the originator of the information is responsible for the input and has the information available interactively and on demand, this is a decided advantage in many applications. The third reason for doing administrative applications in APL is the power of the language itself. Many of the applications that we are presently running were programed in approximately one-third to one-tenth the time that it might have taken us to program the same application in another language.

One of our most dramatic examples of this happened just within the past month. We were approached by a school in our area and asked for a on-line registration system. When the registrar was asked when this registration system was needed, she responded, "next Tuesday", and most of the people at the meeting laughed. The registrar was asked if she could sit down that afternoon and with an experienced programmer outline exactly what she needed, which she did. At the end of Friday afternoon a flow chart and written description of the specifications were completed. The student file and the course file were available in machine readible form and were delivered to the SECOS computer center on Monday afternoon. Some of the programming was completed before receiving the data on Monday afternoon, and the rest of it was completed late into the night Monday night. The program was up and running Tuesday morning. Additional features have been added to the program since then. However, through the power of the language, we were able to produce a program within three days which served the immediate needs. Certainly that level of power and the ability to respond to people's needs will mean that our list of administrative programs in APL will continue to grow at an excelerating pace over the next several years.

* The programs listed above were developed at Cornell Medical School in New York City under the direction of doctors Hagamen and Weber.

AN ON-LINE MIS TO ASSIST THE MANAGEMENT OF FACILITIES

S. C. Fallis
D. D. McElroy
E. E. Payne
Planning and Management Systems Division
University of Texas at Dallas

AN ON-LINE MIS
TO ASSIST THE MANAGEMENT OF FACILITIES

by

S. C. Fallis, D. D. McElroy, and E. E. Payne

I. INTRODUCTION

Building space is a valuable resource that should be managed. At the University of Texas at Dallas a facilities management system has been implemented to assist in the allocation and planning of space.

The allocation of space. The allocation procedure is based upon formulas which use empirically developed standards for space requirements. Each budgetary unit is allocated space according to "formula" and "special request." Formula space is the estimated standard square footage that an organization should occupy. A special request may be made in situations where an administrator feels that the formula does not provide for certain special requirements of his organization. These special requests must be justified (no justification is required for formula space). Approval of a special request is dependent upon the strength of the justification and the availability of space. Normally, all formula space needs are filled before any special requests are considered.

The planning of space. The planning for future space requirements is facilitated by the development and acceptance of reasonable quantitative standards, and by the availability of a system that can produce the answers to "what if" questions concerning space requirements.

The management system. The formalized methods and procedures that are used in the allocation of space, and in the planning of space, are a management system.. A significant and integral part of this management system is a special purpose on-line management information system (MIS). The MIS supports the entire management system (methods and procedures) by supplying information.

II. Objectives of the Management System for the Allocation and Planning of Space

The University of Texas at Dallas is a developing institution. It was originally a research foundation with teaching activities only in graduate studies. In 1969 the scope of the institution was expanded to include upper level undergraduate students. The University is now in the midst of a large building proram, preparing to open for 4,000 undergraduates in the Fall of 1975. Therefore, the management system was developed not only to manage the present space, but also to assist in the planning of new space. The objectives of the management system can be summarized as:

1. Provide a procedure to assure an efficient allocation of space to the present operating organizations.
2. Provide a continuously updated inventory of available space and the assignment of space.
3. Provide the ability to determine the standard (formula) space requirements of any organization based upon: (a) function the organization performs; (b) staffing level; and (c) equipment.
4. Provide the ability to determine the effects of altering the space formulas (standards).
5. Provide the ability to determine the standard space requirements for adding new functions, such as new teaching programs.

III. Space Allocation

Space is allocated to budgetary units (organizations). The basis of the allocation is a set of space formulas (standards). The formulas were empirically developed. They serve as criteria for evaluating allocations. The "formula" space alone is adequate for about 75 percent of the organizations. In cases where the formula space is not adequate "special requests" for additional space are granted. The formula space is based upon such parameters as personnel, semester credit hours, and equipment.

Space is normally allocated once a year as a part of the annual budgeting planning cycle, after the funding levels have been determined for each organization. Data from the budget (personnel, semester credit hours, equipment,...) are necessary input to the space allocation formulas.

Space may also be allocated on an ad hoc basis at any time during the year, to handle un-anticipated situations. Such an allocation is normally initiated by the organization that has encountered the special need.

IV. Overview of the Allocation Procedure

Figure 1 illustrates the general flow of the procedure for the allocation of space. The procedure is basically the same, both for normal allocations made during the annual budgeting/planning cycle, and for ad hoc requests made during the year.

The Office of Institutional Research furnishes technical assistance in preparing the request. After the request goes to the Budget Council, the Office of Institutional Research is again available to furnish any technical assistance desired.

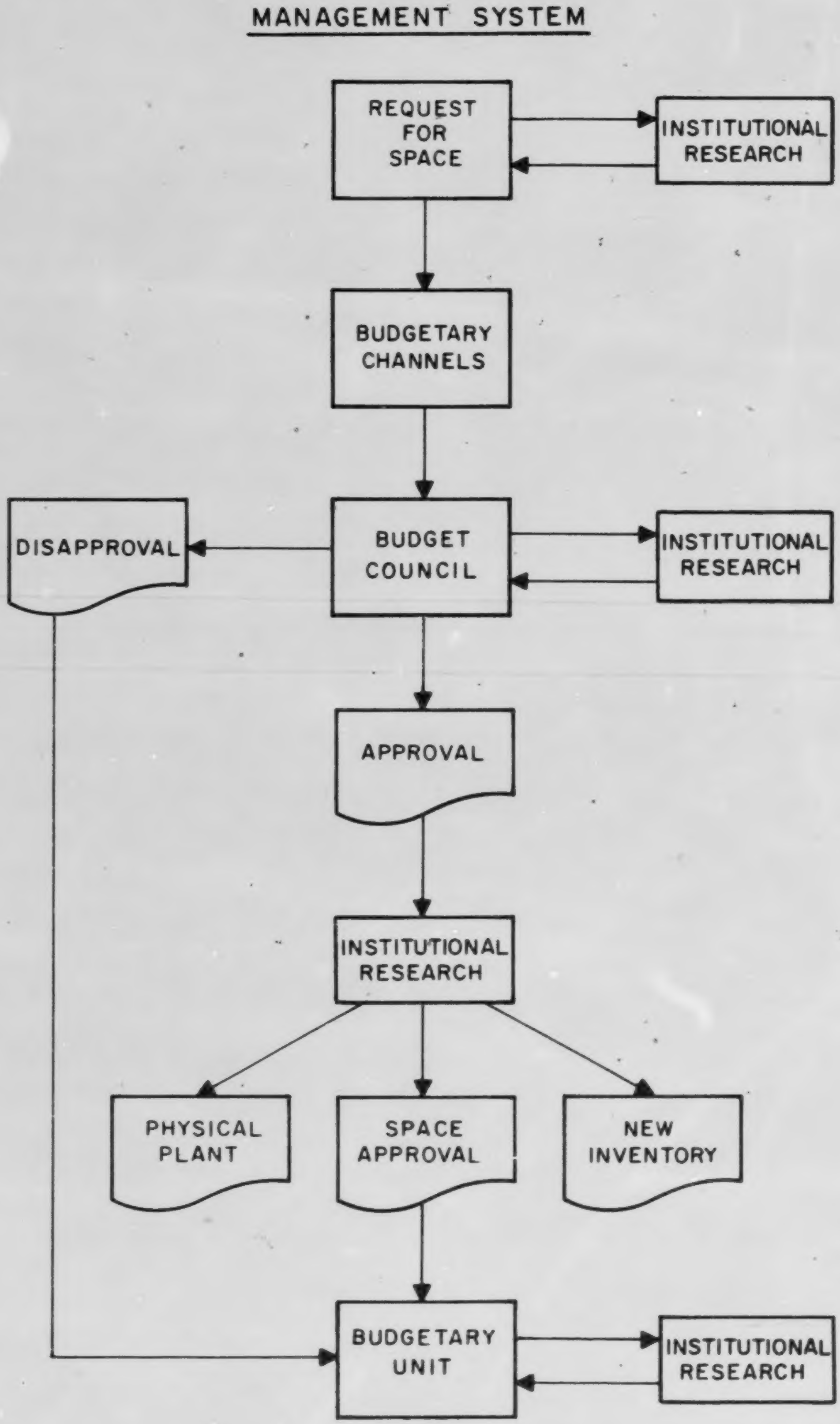

MANAGEMENT SYSTEM
REQUEST FOR SPACE
INSTITUTIONAL RESEARCH
BUDGETARY CHANNELS
DISAPPROVAL
BUDGET COUNCIL
INSTITUTIONAL RESEARCH
APPROVAL
INSTITUTIONAL RESEARCH
PHYSICAL PLANT
SPACE APPROVAL
NEW INVENTORY
BUDGETARY UNIT
INSTITUTIONAL RESEARCH

Figure 1: Overview of the Space Allocation Procedure

the Facilities Management Subsystem consists of five workspaces and a file, referred to as the Data File. The Data File is used to pass data between the work spaces. The Inventory File is used also for obtaining current room data. The subsystem can be used either to generate a complete "Space Report" (as done during the budget/space planning cycle), or to generate only one or two reports at a time.

VI. Facilities Management Subsystem

Figure 2 outlines the structure and activity of the Facilities Management Subsystem. The top section indicates file activity to and from each work space. The middle section details the work spaces and the bottom section indicates reports that are generated as output from each work space.

The first work space, RP1, receives and stores in the Data File the formula numbers and parameter values. It has the capability of producing reports that explain the formula standards and parameter values (used as an appendix in the total Space Report).

The second work space, RP2, reads the formula and parameter values from the Data File, and then generates the formula calculations for each organization. This formula amount is, in turn, stored back in the Data File. A report is generated for each organization showing the parameter values used and total formula criteria for that organization.

The third work space, and probably the most used, calculates assigned space on the basis of the Inventory File and stores this data in the Data File. This work space is used during the year in conjunction with the update procedure. The report produced by this section is an individual inventory of currently assigned space for each organizational unit. This report serves as the official notification of any change in space assignments.

The approval of the request for space is sent to the Office of Institutional Research where the Facilities Inventory is updated. No assignment of space is official until the inventory is updated. A report of the approved assignment of space is generated by the MIS. A copy is sent to the budgetary unit which made the request, and to the Physical Plant. Other organizations that are affected are also notified of the change.

V. Overview of the MIS

For the management system (for space allocation and planning) to perform successfully requires rapid and accurate processing of large amounts of data. Therefore, to provide the necessary information to management, on the required timely basis, an on-line management information system was developed.

The first section of the MIS developed was the Room Inventory subsystem. This subsystem provides data on each room, including such facts as building, room number, dimensions, capacity, room type, room usage, budgetary unit assignment. The subsystem allows a typist to easily enter all room data from a terminal keyboard. The square footage is automatically calculated from the room dimensions.

In conjunction with establishing the Room Inventory, an updating feature was developed. This feature facilitates the alteration of any data in the Inventory through an on-line terminal. To make a change, the subsystem displays all the current information about a given room and then changes need only be typed under the old data. During the first year of operation the inventory files have been maintained by one person working less than quarter time.

Another part of the MIS system, referred to as the Facilities Management Subsystem, produces most of the management level information. Structurally,

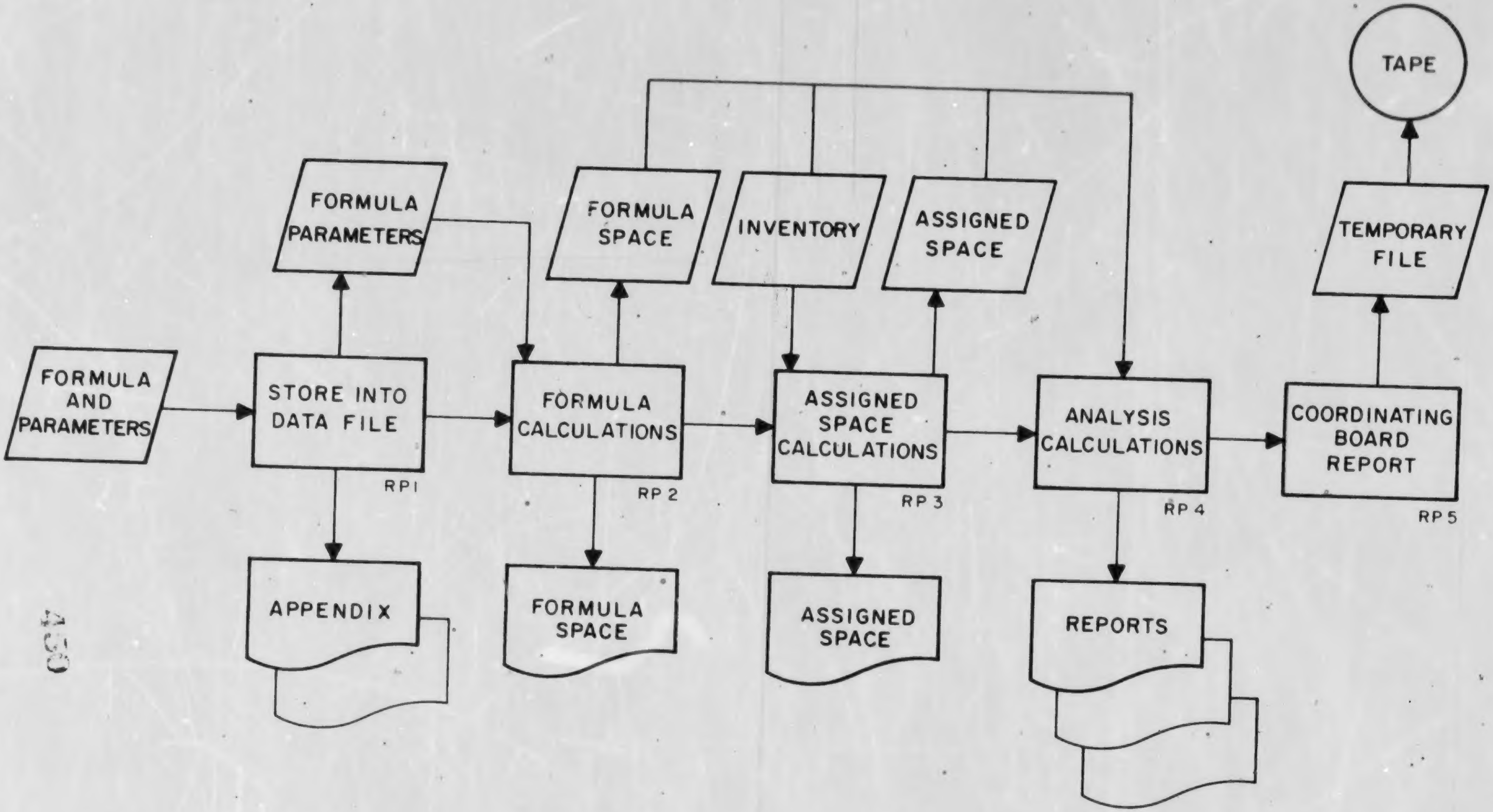

FORMULA AND PARAMETERS
STORE INTO DATA FILE
RP1
FORMULA PARAMETERS
APPENDIX
FORMULA CALCULATIONS
RP 2
FORMULA SPACE
FORMULA SPACE
INVENTORY
ASSIGNED SPACE CALCULATIONS
RP 3
ASSIGNED SPACE
ASSIGNED SPACE
ANALYSIS CALCULATIONS
RP 4
REPORTS
COORDINATING BOARD REPORT
RP 5
TEMPORARY FILE
TAPE

Figure 2: Facilities Management Subsystem

The fourth work space, RP4, reads data from the Data File and Inventory, and prints out various analyses. One such report compares total assigned space versus "formula" criteria and gives their difference and ratio. There are other reports that show analyses such as space by building, and space by usage.

The final work space creates a required report for the Texas State Coordinating Board. This report is actually put into a temporary file, then on tape, and then the data are converted to a form compatible with the State system.

VII. Conclusion

The MIS is not a completed, finished product. It is instead a flexible tool which is modified, sometimes daily, as new management demands arise. APL has helped greatly to make this concept successful.

AN APL SOLUTION TO A DATA PROCESSING PROBLEM

Robert E. Schaulis
Director of Information Services
Coast Community College District
Costa Mesa, California

AN APL SOLUTION TO A DATA PROCESSING PROBLEM

Robert E. Schaulis
Director of Information Services
Coast Community College District
Costa Mesa, California

ABSTRACT

The Coast Community College district (Orange Coast College and Golden West College) has had extensive experience with APL. Starting in the 1968-69 school year and continuing today, APL is the only interactive terminal based language used by the district's colleges.

Although by district policy use of APL is limited to CAI, a number of administrative applications have turned up on the system. These include an application in the fiscal area and a general secretarial system, which will be described in some detail.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

INTERACTIVE ANALYSIS OF STUDENT RECORDS

P. C. Berry
APL Education
IBM Scientific Center
Philadelphia, Pennsylvania

R. M. McCarthy
Research and Technical Services
Computing Center
SUNY/Binghamton

INTERACTIVE ANALYSIS OF STUDENT RECORDS

P. C. Berry
Manager of APL Education
IBM Scientific Center
Philadelphia

P. M. McCarthy
Research and Technical Services
Computing Center
State University of New York
Binghamton

ABSTRACT

Statistical analyses of student records are being obtained from an experimental package of APL programs. These provide the institutional research office with the ability to respond to novel requests in a matter of minutes, where formerly these required them to write and then execute a batch program for each new request.

The file itself continues to be generated and maintained by COBOL programs operating in batch mode, using standard machine-readable records.

The package makes use of a file description to translate back and forth between the codes actually stored in the file and their English descriptions. This permits the user to write his requests using familiar terms or abbreviations, and permits the program to generate a report in which the various entries are appropriately labelled. The description consists of a set of tables showing the names and headings used for each field, and, within a field, the permissible codes and the meaning of each. Simultaneous cross-tabulations of any number of fields may be requested. The resulting tables appear in the APL workspace, where they may be subjected to further computation, or labelled and printed, as desired.

A mechanism is also provided for searching the file for records meeting combinations of criteria, and for displaying and editing individual records. The search technique makes use of a program that compiles propositions stated in the English-like user language into APL statements which are then executed during the search.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

SECURITY FACILITIES IN AN APL SYSTEM

Joseph W. Fries Richard Cox

APL Services, Incorporated
Houston, Texas

SECURITY FACILITIES IN AN APL SYSTEM

I. INTRODUCTION

The purpose of this paper is to illustrate the framework of internal and user-designated security provisions in the APL Time-Sharing System. With the advent of "computer theft," confidential data bases, executive espionage, etc., the computer-based community is becoming more and more conscious of the need for stronger, more effective security controls for their installations.

The system safeguards referred to in this paper are those which are incorporated in the ACTION/APL Time-Sharing System. Although they are by no means foolproof, since no system really is, they are providing an effective measure of security, both at the system level and at the user level. The hardware being utilized in the ACTION/APL System is an IBM 370/155 computer, under the control of an OS/HASP host system.

Security controls can be reduced to two basic concepts in a time-sharing environment: system-oriented and user-oriented. Both concepts are of equal importance, and must be concurrently developed and controlled, to insure maximum effectiveness of system security measures.

II. APL SECURITY - SYSTEM ORIENTED

System operator . . .

The APL system operator is the primary link between the computer operations staff and the remote user. Through him, all users' requests are monitored and directed to the proper persons. However, one of the main responsibilities of a system operator must be that of system security. Occupying one of the most powerful positions in the time-sharing network, the operator has:

- full control of system access - All users' sign-on messages are recorded at the operator's terminal, allowing him a constant map of the entire system utilization. If unauthorized persons do gain access to the system, he has the power to bounce and/or lock those persons off of the system until they are properly identified.
- access to user-designated locks of user numbers, programs and workspaces.

As the above examples illustrate, the efficiency and security of a time-sharing system are heavily dependent upon the abilities and performances of the system operator.

System control . . .

Unfortunately, over the past few years, internal system security has been de-emphasized, with the main thrust of development being directed toward system speed. In-house installations are not as directly affected by this decline in security development, as are those in the time-sharing community. Thus, more and more protection must be afforded the time-sharing users by the time-sharing vendors.

User validation and access authorization to ACTION/APL is maintained through the use of a 7-digit authorized access number and a user designated lock. The sign-on procedure consists of the following format:

```
)1234567:LOCK
```

where ")" is a system recognition symbol for all system commands, "1234567" is the authorized access number assigned to that particular user, and "LOCK" is the user-designated lock. A directory of all authorized numbers and their associated locks is internally maintained, and all user sign-on messages are compared to this directory to insure their inclusion in it. If the sign-on message does not exactly match the directory entry, the system will respond with one of the following messages:

```
INCORRECT SIGN-ON
NUMBER NOT IN SYSTEM
```

Thus, entry to the system is literally impossible if an access number is not included in the system directory. (More on this later *)

Together with authorized access, measures must be taken to insure the protection and integrity of all users' data that is stored in the system. Although time intervals may vary, all time-sharing systems must adhere to one or more of the following procedures:

- a daily system back-up tape of all customer workspaces and files.
- end-of-week dumps of all customer workspaces and files.
- end-of-month dumps of all customer workspaces and files.

These back-up tapes are essential for two reasons:

1. In the event that a customer destroys any stored data or accidently drops or erases any files or workspaces, the back-up tapes will provide the most recent version of the users' data for purposes of restoration.
2. In the event of a system failure or hardware crash, the maximum data loss a user may suffer is back to the most recent back-up tape, or a maximum of 24 hours.

*User locks are internally secured by the use of a "hashing" system. Hashing is a procedure whereby user-designated locks are internally converted, by using a non-meaningful algorithm:

EXAMPLE: a lock such as "HELLO" might be converted to "A6R"

If a person should gain access to the system directory, and produce a master listing of that directory, the "A26R" lock would appear on the printout. Only by using the hashing algorithm, which is an external software control, would he be able to gain access to a user's lock.

III. APL SECURITY - USER ORIENTED

Locks and access numbers . . .

In a time-sharing environment, user-controlled security measures are of the utmost importance. Each user must be unique, that is, he must have a certain characteristic which is applicable to him and him only. This is accomplished through the assignment of a system-authorized access number, which acts both as a security measure and an identification for internal accounting. Upon validation, a lock is assigned to each user number, for purposes of initialization:

EXAMPLE: 1122334:SAO

Through the use of this number and associated lock, a double measure of security is afforded the user upon initial use of the system. One of the greatest and most assuring safeguards afforded to users is the ability to change these locks, at will. This is done upon the normal system sign-off procedure, in the following manner:

)OFF:NEWLOCK

The frequency of lock changes is fully dependent upon the opinion of the individual user, and may consist of items such as the user's initials, birth dates, etc. To allow further security measures at the user level, all programs and/or workspaces may also be locked by individual users. To lock a workspace, one can simply assign a password, in the following manner:

WORKSPACE:LOCK

where "WORKSPACE" is the workspace to be locked, and "LOCK" is the password to be assigned.

To lock a function within a workspace, the following format is used:

FUNCTION

(The operator is made with , backspace .) When all of these measures are combined, they provide for four separate levels of security available to all users of ACTION/APL.

File and data base security . . .

There has been a recent trend to the use of proprietary data bases and shared access of large data files. Access to data bases such as the CAB FORM 41 and the

NBER is system-controlled, through the use of specially-designated user numbers. However, each user is afforded the capability of privileging other users to read, write, or read/write to any or all of his files. This is done through the reaction of an access matrix, which contain those user numbers privileged to access the owner's files. Additions and/or deletions to the access matrix are under the full control of the owner of the particular file, and may be made at will.

IV. SUMMARY

In conclusion, it must be noted that many steps are now being taken to further refine effective security measures in computer installations. Some of the methods currently being investigated are:

- the MULTICS system, which incorporates an internal system-programming check to ensure that only authorized persons have access to internal system controls.
- palm-reading minicomputers and programmable locks, to control authorized access to computer installations.
- security consulting groups, offering their services to the installation to test existing security systems, and developing better safeguards.
- tighter controls, especially at the Federal

level, to insure the integrity and protection of confidential data bases, such as those maintained by the IRS and the FBI.

Security is the responsibility of every individual within an organization, and it will be effective only if it is enforced by all of those individuals.

FEATURE ANALYSIS: STUDENT RECORDS SYSTEMS

Verner R. Ekstrom
Assistant Provost
University of Oklahoma

FEATURE ANALYSIS: STUDENT RECORDS SYSTEM

A presentation by Verner R. Ekstrom, Assistant Provost for Academic Records, University of Oklahoma, at the College and University Systems Exchange National Conference, New Orleans, Louisiana, December 11, 1973.

Introduction. This presentation will cover the systems problems we faced in the student's record area, a brief description of the search for a resolution of those problems, the decision that was made, and the standards that were established for the resolution of the problems. We will also discuss the specification and design of the proposed new system to be purchased and installed. Towards this end we will present a user definition of the 19 applications to be installed along with a statement of potential benefits to be derived from each of those application areas. We hope the definition and benefit statement will be of value to those institutions approaching the development of a real time system in their student records area.

In the fall of 1973 the enrollment on the Norman campus of the University was 19,647 students. In addition there were 1,500 students registered in various programs at the Health Sciences Center and over 60,000 registered in our vast programs in Continuing Education and Public Service throughout the State of Oklahoma and the world. It has been the announced goal of the institution to integrate all of our student records into one system.

The Problem. Several years ago we inherited a computer admissions system, a student master file system, and a class schedule file system that had tremendous problems of both poor design and horrible execution. Our first efforts were devoted to eliminating the execution problems. Many, many of the programs were redesigned and reprogrammed and many controls and edits were implemented. Nevertheless, we were still left with the poor design of the overall system. Additionally we realized that the system was not responsive to our needs to integrate the records of the extensive programs in Continuing Education and at the Health Sciences Center. A solution to these difficulties was sought.

The Search. In searching for an adequate solution to our difficulties many institutions throughout the U.S. were visited. Several vendors were consulted about the possibility of buying software packages. Finally two attractive alternatives arose for our consideration. One of these was to build or buy a more adequate sectioning algorithm since our registration pattern was one of the key difficulties that was perceived by deans, department heads, faculty and students. The other was to proceed to an on-line system such as the ones installed at Central State University, Edmond, Oklahoma, Regis College, Denver, Stanford, Brigham Young, Harper College, Palatine, Illinois, Johnston County Community College, Kansas, and New Mexico State University, Las Cruces. All of these systems were examined in some detail.

The Decision. A relatively small but effective software vendor made us an offer we could not refuse. They proposed to install and allow us to evaluate their real time system at the University for a very modest fee. It was also determined that the cost of the implementation of their system was very attractive. We decided to accept their offer.

Performance Standards. In contracting with the vendor we felt it was very important to have an adequate set of performance standards. These standards were jointly developed by the software vendor, the offices of Admissions and Records and Administrative Data Services. A key clause in our contract was that satisfaction was guaranteed, i.e., "Based upon any criteria it may choose the University of Oklahoma may .... reject the system without further obligation".

Not content with this simple guarantee, however, the performance standards included a specification of the benefits to be derived from the system, a description of the system as installed elsewhere, a list of our batch output requirements, a specification for the terminal screen displays, a list of the minimum data elements to be incorporated in the system, a description of the report generator capability, a stipulation that thorough training would be provided, a commitment to providing both user and system documentation, a statement as to the space and other support required by the vendor, and a statement on the personnel to be committed by the vendor, Admissions and Records and Administrative Data Services. The performance standards were included as an addendum to our contract.

The systems description was stated to be a minimum one. It basically described the system as installed at another major university. After we selected the terminals to be used, however, we found that the new terminals would require a fairly major change in programs. Additionally, as we began to examine our particular needs we found that our requirements were considerably more extensive than the system could provide. Consequently, we decided to redesign the data base and all of the application programs. This also required some modifications in the executive monitor and the report generator capabilities of the system.

The Application Areas. A major objective of this paper is to present the systems specification as seen by the user, i.e., the admissions and records personnel. A representation of many of the CRT terminal screens is available from CAUSE upon request; a discussion of the application and benefits derived is included here as Appendix B.

The bulk of our presentation at the CAUSE meeting was devoted to an examination of some of the terminal screens and a discussion of the definition of the system. During the presentation questions were encouraged and discussion about several of the features of the system was lively.

Status. The system as described in Appendix A and B is now installed for evaluation at the University of Oklahoma. The evaluation effort involves the offices of Admissions and Records for Norman, the Health Sciences Center, and Continuing Education. Over 30 other offices are also involved including Administrative Data Services and all of the dean's offices. After evaluation if we decide to purchase the system we will proceed to the final systems definition. There will then be the final evaluation of each of the sub-systems prior to their installation. It is expected that the system will be operational in the fall of 1974.

2/1/74
Appendix B

THE APPLICATION DEFINITIONS AND POTENTIAL BENEFITS TO BE DERIVED FROM THE UNIREC EVALUATION SYSTEM

The following statement defines the applications to be installed at the University of Oklahoma if the University Records (UNIREC) System is purchased from the Computer Application Corporation. The system will completely replace all computer programs and procedures related thereto now existent in the Admissions and Records area including those which provide services to other offices.

Key Assumptions

If the UNIREC System is installed it is expected that all capabilities listed herein will be available. In identifying these benefits, however, the following assumptions were made:

1. That Cathode Ray Tube (CRT) and other terminals will be installed in each office shown in the "UNIREC Terminal Configuration Plan" (See Page 5).
2. That personnel resources necessary to conduct the operations proposed in each application area will be available.
3. That University policies and procedures will be adjusted as needed to accommodate the requirements of the system as defined herein.
4. That computer time for first shift on-line processing and overnight batch processing will be available as needed.

Additionally we have made the following procedural assumptions:

1. That student records will be accumulated on the system from the day the system is implemented, i.e., we will not make a major effort to enter earlier data on students.
2. That students will be given a complete transcript of his records rather than simply a semester to semester grade report.
3. That control of class enrollment after classes begin will remain with Admissions and Records (presently it is transferred to departments).

It should be noted that all procedures defined herein are subject to review and change during our evaluation effort, i.e., alternatives to our assumption or to our definition will be considered during evaluation.

Policy Questions

We believe that installation of the UNIREC System will also raise certain policy questions and problems that must be resolved. Key among these will be the question of authority and responsibility for student data. We propose that the admissions office retain ultimate responsibility for the creation, maintenance and reporting of all data maintained on the UNIREC Data Base but that this responsibility be delegated along with commensurate authority as appropriate to other offices and as mutually agreed upon by all parties.

Additionally we point to the following policy questions:

1. Who will pay for the installation and operational costs of terminals at each location?
2. Who will pay for special reports specified by users at various terminals throughout the system?
3. What will be the impact of the new system on our policy on the release of student information?

Answers to these questions will need to be given before the system is installed.

Potential Benefits

Following each applicant area, we have listed statements on the potential benefits which are among these expected from the UNIREC System. These statements are somewhat speculative in nature, but it is expected that these benefits will largely, if not entirely, accrue to us if the system is installed as defined and if the departments, colleges and offices implement them. On page 6 a summary of the potential benefits is arranged in tabular form.

It is impossible to quantitatively measure in any meaningful way all of the benefits to accrue on the overall quality of a system; however, some means or guidelines for assessing the expected outcome should be available. These statements provide these guidelines. The list of benefits include only those that are over and above the benefits now obtained from the present Student Information System.

Benefits have been categorized under twenty-five headings as follows:

Potential Benefits

Services:

1. Improved Student Services
2. Improved Faculty Services
3. Improved Staff Services
4. Improve Instruction
5. Improved Schedule and Course Offerings

Planning:

6. Improved Planning
7. Improved Reporting
8. Facilitate Special Reports
9. Improved Report Scheduling

Control:

10. Improved Control
11. Improved Monitoring
12. Orderly Growth
13. Improved Record Security
14. Improved Utilization of Resources
15. Localize Responsibility
16. Extend System Familiarization

Costs:

17. Reduced Costs
18. Less Clerical Effort
19. Avoidance of Redundancy
20. Paper Reduction

Data Base:

21. Expanded Data Base
22. Improved Accuracy
23. Improved Access
24. Improved Speed of Response
25. Improved Timeliness and Currency

Sometimes it has been very difficult to separate the potential benefits into these categories. In such cases we have included the benefits under all applicable categories with the emphasis changed to reflect the emphasis of the category. For example, under the application "Academic Resources - Course" we have listed the system capability of computer preparation of a State Regents report under both the "Improved Reporting" and the "Less Clerical Effort" benefit categories with appropriate change in emphasis. This introduces some unavoidable redundancy in our benefits statements. Though we have not included a category on "increased costs" it should be noted that the achievement of many benefits will essentially increase costs. For example, because of costs, we discontinued the routine preparation of advisory transcripts some years ago. It would be naive to assume that the new system will produce this product without new costs (perhaps at a substantially lower cost than former methods however). As another example our present system provides little or no support for maintenance of student degree plans - the UNIREC system does. However, were this capability to be employed it will take considerable effort to enter and maintain the data.

Alternatives

There are many alternatives to the systems definitions contained herein. Many of these alternatives will be considered during our evaluation efforts - in fact - identification of alternative procedures is an important goal of our evaluation. The Admissions and Records staff has identified several important alternatives. In the paragraphs that follow we will stipulate a few of these.

Alternative 1 - Optical Reading of Data

If Optical Mark Scanning or Optical Character Recognition (OCR) equipment were to be secured by the University it will create a potential for automatically entering data in a number of areas. For example, it would be possible for students to record their course requests on optical scan sheets and obviate the need to keypunch this data. Grade lists could be printed on OCR sheets on which instructors could enter grades. The grades could then be mechanically read rather than keyed by terminal operators.

Alternative 2 - Registration and Advising

Several alternatives to the registration and advising definition included herein have been discussed. At least one alternative will be presented during the evaluation effort.

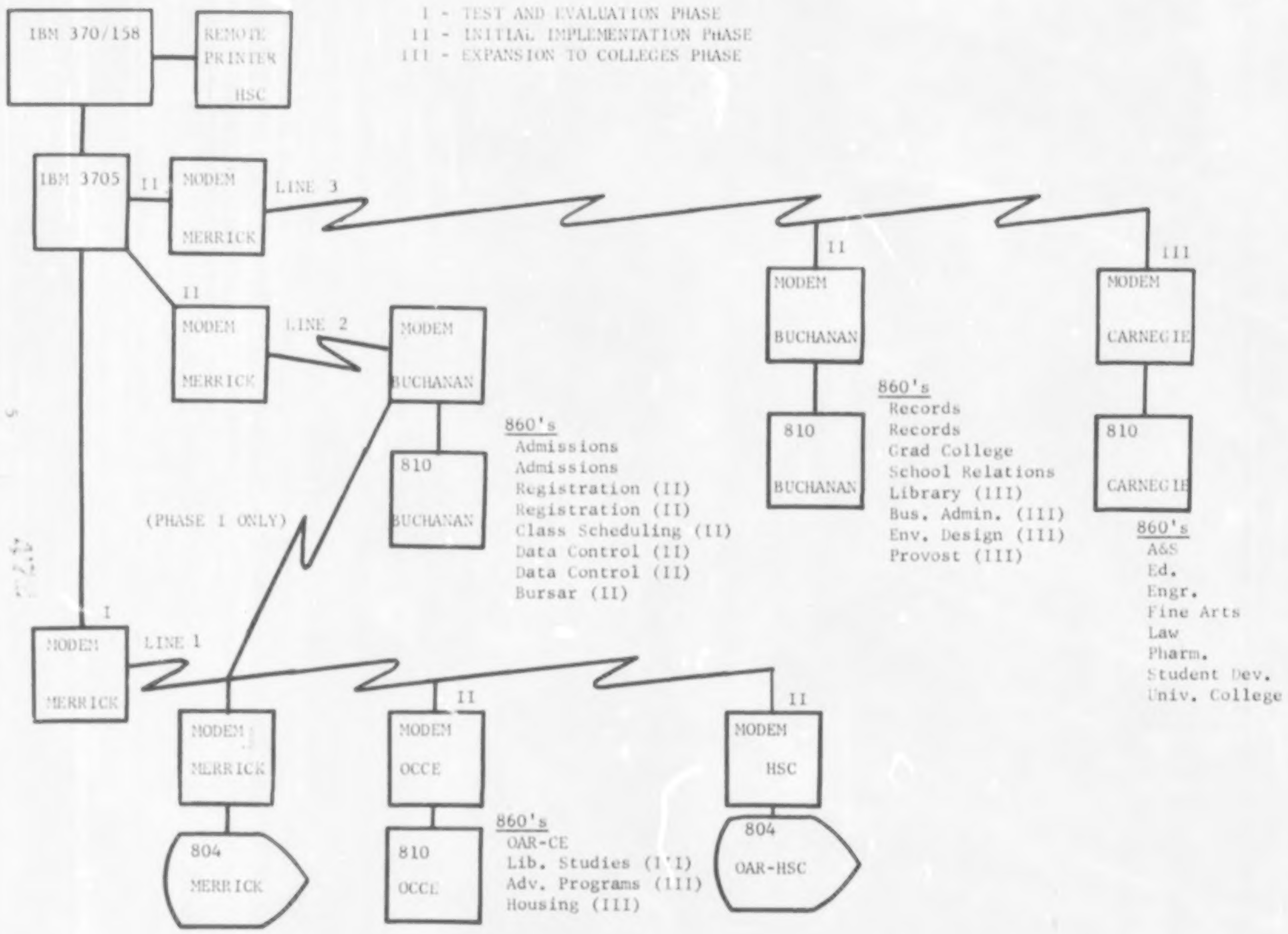
Acknowledgment

We wish to express our appreciation to the UNIREC task force for their assistance in the preparation of this document. The task force included Mr. Wayne E. Reynolds, Chief Consultant, Computer Applications Corporation, Mr. Charles Rice, Mr. Harry Blaylock and Mrs. Maisy Curry of the Administrative Data Services staff, and Miss Maureen Murphy, Mr. Paul Bolen, Mrs. Racheal Keely, Dr. Ronald Stafford, Mrs. Gretchen Lowsley, and Mrs. Lola Wilson of the Admissions and Records office. Without their assistance this document could not have materialized.

Verner R. Ekstrom

UNIREC TERMINAL CONFIGURATION PLAN - TENTATIVE

2/1/74


I - TEST AND EVALUATION PHASE
II - INITIAL IMPLEMENTATION PHASE
III - EXPANSION TO COLLEGES PHASE
IBM 370/158
REMOTE PRINTER HSC
IBM 3705
II
MODEM MERRICK
LINE 3
II
MODEM MERRICK
LINE 2
MODEM BUCHANAN
810 BUCHANAN
860's
Admissions
Admissions
Registration (II)
Registration (II)
Class Scheduling (II)
Data Control (II)
Data Control (II)
Bursar (II)
(PHASE I ONLY)
II
MODEM BUCHANAN
810 BUCHANAN
860's
Records
Records
Grad College
School Relations
Library (III)
Bus. Admin. (III)
Env. Design (III)
Provost (III)
III
MODEM CARNEGIE
810 CARNEGIE
860's
A&S
Ed.
Engr.
Fine Arts
Law
Pharm.
Student Dev.
Univ. College
I
MODEM MERRICK
LINE 1
MODEM MERRICK
804 MERRICK
II
MODEM OCCE
810 OCCE
860's
OAR-CE
Lib. Studies (III)
Adv. Programs (III)
Housing (III)
II
MODEM HSC
804 OAR-HSC

UNIREC APPLICATION BENEFITS MATRIX

Operation Benefits	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S	T	U
Services:																			
1. Improved Student Services	X	X	X					X	X	X	X	X	X		X	X			
2. Improved Faculty Services			X		X	X		X				X	X						
3. Improved Staff Services	X		X	X		X	X	X		X	X	X	X	X	X			X	
4. Improved Instruction				X				X						X	X				
5. Improved Offerings				X			X	X			X				X				
Planning:																			
6. Improved Planning		X		X			X	X		X	X		X	X			X		
7. Improved Reporting				X	X		X	X		X				X	X				
8. Special Reports										X				X					
9. Improved Scheduling											X					X			
Control:																			
10. Improved Control	X	X		X			X	X			X				X	X	X		X
11. Improved Monitoring		X						X			X		X			X			
12. Orderly Growth		X		X				X					X		X			X	
13. Improved Record Security		X													X	X			X
14. Improved Utilization				X		X	X								X	X		X	
15. Localize Responsibility	X	X													X	X	X		
16. Extend Familiarization	X	X	X										X		X	X	X		
Costs:																			
17. Reduced Costs	X	X	X	X	X			X		X					X				
18. Less Clerical Effort	X	X	X	X	X		X	X	X	X	X				X	X	X	X	
19. Avoid Redundancy	X	X		X	X		X	X		X					X		X		
20. Paper Reduction	X	X	X	X	X			X						X		X	X	X	
Data Base:																			
21. Expanded Data Base	X	X		X			X			X			X	X					
22. Improved Accuracy	X	X	X	X		X		X	X	X				X		X			
23. Improved Access	X	X	X					X		X				X	X	X	X		
24. Improved Response	X						X	X						X	X	X			
25. Improved Timeliness							X	X		X									

Application Codes

A School and College Relations
B Admissions
C Transfer Evaluations
D Academic Resources - Facilities
E Academic Resources - Courses
F Academic Resources - Instructors
G Academic Resources - Class Schedules
H Registration, Advising and Schedule Changes
J Student Billing
K Student Records
L Grade Reporting
M Advisory and Official Transcripts
N Degree Planning
P Statistics and Institutional Research
Q Integration of Student Records
R System Operations and Control
S System Documentation
T System Program and Maintenance
U System Security

Application: A - School and College Relations

Definition: Contacts by mail, telephone and personal visits in schools and colleges are recorded in either a prospect, applicant, or student file. Labels are produced for mailing of recruiting publications, bulletins and applications. Records of contacts and publications sent are kept on a flexible "credentials checklist" file. As new contacts are made, the file is referenced to ascertain the nature and extent of earlier contacts. Contact and publication records are purged automatically after aging.

As test scores are received, students with unusually high scores are sent special letters of encouragement. Recruiting lists are prepared as needed. Prospect records are moved to applicant files as application for admission or ACT test scores are received.

Potential Benefits:

1. Improved Student Services: Response time to prospect and applicant inquiries by letters and calls will be reduced. Confusion because of lack of coordination of responses by School and College Relations, Admission, Housing, etc., will also be reduced.

3. Improved Staff Services. Recruiting lists will materially assist School and College Relations efforts. Immediate and direct access to applicant and student records will also aid those offices in their operation.

10. Improved Control. Coordination of responses will result in better control of the prospect and applicant's record.

15. Localize Responsibility. School and College Relations will control all data on prospects and will be able to improve data base on applicants.

16. Extend System Familiarization. All procedures relating to the School and College Relations Office handling of prospect's records will be observable by admissions. Additionally, admissions responses to inquiries by prospects will be available to School and College Relations. Similar sharing of information in handling of applicants will be possible.

17. Reduced Costs. Cost of recording contacts will be reduced. Cost of distribution of publication will be reduced. Cost of preparation of special letters of encouragement will be reduced.

18. Less Clerical Effort. Work involved in distribution of bulletins and other publications will be reduced. Filing of some contact records will be eliminated. Multiple keying of some prospect-applicant data will be eliminated.

19. Avoidance of Redundancy. Separate record systems in School and College Relations and Admissions will be unnecessary. Multiple keying of data maintained by these offices will be eliminated.

20. Paper Reduction. Need for paper reports to School and College Relations on applicant status or prospect test scores will be reduced.

21. Expanded Data Base. New ACT data and data on prospect contacts and publications sent will be available to all offices on the system.

Application: A - School and College Relations

Benefits: (cont'd)

22. Improved Accuracy. Echo checking of data entered will improve accuracy of prospect and applicant records. Simple keying of data or automatic entry from ACT test tapes and subsequent verification at the time of later contacts will also improve accuracy of data collected.

23. Improved Access to Student Data. School and College Relations and college offices with terminals will have access to prospect and applicant data directly without hard-copy printouts.

24. Improved Speed of Response. Inquiries as to status of prospect or applicant will be answered without resorting to telephone calls or memos.

Application: B - Admissions

Definition: As applications are received, the data thereon are entered into the system. If a prospect record exists, it is moved to the applicant file and additional data is added. Paper documents are logged on the system for retrieval from microfilm records as the paper documents are aged and destroyed. Test scores and other data from ACT are received on magnetic tape and read directly to the applicant file. Reference to data on the ACT tape not added to the system is maintained in "credential checklist" records.

The system makes preliminary recommendations (not final judgments) concerning admitting the student, based on programmable admission criteria. Notices of missing information are prepared prior to the final admission decision. Certificates of Admission and Verification or Rejection Notices are produced. As they are returned by the applicant, verification forms with edited and new data are used to update files. Applicant files are monitored to insure timely response in handling of the applicant's record. Periodic detail and summary statistical reports are prepared.

Graduate and professional applicants are handled separately and distinctly from undergraduate applicants. Special records and files relating to these students are maintained.

Applicant records are automatically moved to student files when the student first registers. Records for applicants who do not register are aged, tabulated, and eventually purged from the files.

Potential Benefits:

1. Improved Student Services. We are developing a cadre of records counselors who will serve as the major interface between our students and the University's record system. These counselors will be able to handle all admissions, registration, transcripts, transfer evaluation, degree application, fee assessment, residency, verterans certification, and other records problems. It will be possible for the student to handle all necessary records transactions at one place, through this records counselor. The records counselors will have at their disposal our stenographic pool for personalized correspondence. Additionally, the student's record may be assessed by terminal by the records counselor, making it possible to provide a rapid and integrated service to the student.

6. Improved Planning. Improved data base and reporting will permit better estimates of new student registration projections and analysis of admission decisions.

10. Improved Control. Status of admission will always be known, even at peaks. This will avoid confusion, errors, and maintains control over admission policies while treating all applicants fairly.

    Coordination between the Graduate College and Admissions will materially be improved since current status of graduate applicants will be available to both offices. Transition from undergraduate to graduate work will be facilitated.

Application: B - Admissions

Benefits: (cont'd)

19. Avoidance of Redundancy. Data from the admissions system provides a base of data needed by registration, academic records, Graduate College, Housing, Health Center, Student Development, Financial Aids, Deans' offices, and others. The extent to which these other data bases need to be developed and maintained are substantially reduced.

20. Paper Reduction. Many reports now required to keep various offices informed of the status of applicants will be reduced.

21. Expanded Data Base. Much new family, address, test score, demographics, and academic data will be available in machine processable form or for access through terminals. Though much of this data was previously available on paper documents, access to this data was limited because of costs for retrieval or entry into the system. Much of this data will be automatically entered from the ACT tapes for ACT prospects.

22. Improved Accuracy. Applicant editing of the data on our files will materially improve its accuracy. Verification of data previously entered will also occur as letters, applications, transcripts, and other documents are received.

23. Improved Access. Most records on applicants will be accessible at all terminals on the system.

Application: C - Transfer Evaluation

Definition: As transcripts are received for student transfers from other institutions, their receipt is recorded on the files. For transcripts from institutions sending moderate numbers of students to OU, each course and grade on the transcript is keyed into the system through terminals. The system examines the records of previous evaluation decisions and returns a course equivalence or other decision on the screen. If no evaluation has been previously made for the course, the transcript is referred to a transfer evaluator for a decision. Once a decision is made, it is entered into the system for future reference. This evaluation is then added to the student's record.

Evaluation decisions are specific to the transfer institution and the Office of Admissions and Records making the evaluation decision. For transfers from institutions with few students coming to OU, evaluations are prepared manually and only credit and grade-point summaries are entered into the system. They may be entered by term or in total for the transfer institution. The system automatically generates grade-point and grade-point summaries for each term, both at the transfer institution and at OU. These summaries are accumulated into cumulative statistics in transfer alone, at OU alone, and in grand total.

Potential Benefits:

1. Improved Student Services. Evaluation of transfer credit will provide early knowledge of how the student's credits will apply to work at OU.

2. Improved Faculty Services. Prompt and accurate transfer evaluations will assist faculty in their advising responsibilities.

3. Improved Staff Services. Retrieval of transfer evaluation decisions at their terminal will assist College Relations in the execution of their responsibilities.

16. Extend System Familiarization. Evaluation decisions will be available to all offices for examination and review. Concern relative to specific course evaluation may be discussed and perhaps changed.

17. Reduced Costs. Making an evaluation decision often requires a considerable amount of time on the part of highly trained admissions counselors. Once recorded in the system, the computer will match earlier decisions against specific courses keyed by terminal operators from transcripts for new students applying for transfer admission. Since a junior college transfer, for example, may present twenty courses from over 500 courses at college XYZ, our trained professionals may need to evaluate only one or two courses for a particular student. The saving in time and effort on the part of highly trained personnel will be considerable.

18. Less Clerical Effort. Presently our office prepares four copies of the transfer evaluations for each incoming transfer student. Copies are distributed to the student, the college, and the departmental adviser. Only the transfer grade-point average is posted to our computer records. The system provides the ability to post this information directly into the system by keying of this data directly. The data can then be manipulated in a number of ways, including the printing of advisory transcripts for colleges and departments.

Application: C - Transfer Evaluation

Benefits: (cont'd)

20. Paper Reduction. Only two, instead of four, copies of transfer evaluations need be made, one for the student and one for his adviser.

22. Improved Accuracy. Computer retrieval of earlier evaluation decisions will improve consistency and accuracy of the evaluations.

23. Improved Access. College and departmental offices with terminals could have access to this data directly, without hard-copy printouts.

Application: D - Academic Resources - Facilities

Definition: Building and room resources are entered and maintained and are used to produce classroom and class laboratory charts and untilization data based on the Class Schedule offerings. Other facilities' data are entered for facilities management information. Facilities records may be "scanned" on line.

Potential Benefits:

3. Improved Staff Services. Integration of Class Schedule data with facilities management data will make it possible for the Office of Administration and Finance to operate more effectively.

4. Improved Instruction. Better facilities management will improve the availability of rooms and equipments needed for classes at the right time and in the right location. This can make a substantial difference in the effectiveness of the courses being taught.

5. Improved Schedules and Courses Offerings. Where courses require special equipment or facilities, some management information is needed to provide for these courses. If this does not occur, course offerings may be considerably weaker than they need to be because of the lack of such facilities.

6. Improved Planning. Accurate and complete reporting on the untilization of facilities can help in planning for renovation and/or new facilities.

7. Improved Reporting. Accurate and current reporting of current facilities utilization can help in making an assessment for needed additional facilities.

10. Improved Control. Accurate reporting of resources being utilized by various campus programs can help assure proper balance of resource usages. This will help to insure that more important programs are not lacking resources at the expense of lower priority programs.

    The system can be programmed to notify administrators of departments or individuals making excess or unwise use of facilities.

    The system can also help in the allocation and scheduling of facilities. Since the files are on line, access will provide immediate response to requests for additional space, and use of a particular room.

12. Orderly Growth. Better use of space will allow us to utilize available facilities in the best possible manner in order to avoid serious space difficulty. Poor management of these facilities will cause us to pay for new facilites which are not really required.

14. Improved Utilization. The facilities management system can enable more complete use of available facilities and ultimately reduce the number of rooms and the amount of various kinds of equipment required by the University.

17. Reduced Costs. Presently, we are operating two separate computer systems, one for maintenance of class schedule information, the other for maintenance of facilities inventory data. Merging the two into one system will effect economies of operation that will reduce costs in computer operation and in Administration and Finance in securing needed data.

Application: D - Academic Resources - Facilities

Benefits: (cont'd)

18. Less Clerical Effort. Classroom and classroom laboratory charts will be on-line.

19. Avoidance of Redundant Files. Facilities inventory files can be used to obtain data for use in registration and class scheduling, while class schedule data can be used in facilities management. Separate files will be eliminated.

20. Paper Reduction. The need for listing of room charts will be reduced. Paper reports on space inventories will be reduced.

21. Expanded Data Base. The system provices a considerable number of new data elements for coding of facility characteristics and quality.

22. Improved Accuracy. Facilities data will be constantly updated as class scheduling proceeds each semester and as facility renovation and changes are made.

Application: E - Academic Resources - Courses

Definition: A master course record is maintained for all courses entered into the system. The records are controlled by a department and number designation. Cross-listing of courses is identified for ease of retrieval and programmed control of enrollment. If a course is discontinued, a cut-off date is recorded. If key data elements of the record change, the old entry is coded with a cut-off date and a revised course record is created.

Potential Benefits:

2. Improved Faculty Services. An up-to-date catalog of courses will be of value to the Academic Program Council as it deliberates on new courses and programs.

7. Improved Reporting. Each year we must complete an extensive report for the State Regents for Higher Education on all courses offered by OU. This report is now prepared manually. The system will permit us to automatically prepare this report.

17. Reduced Costs. The maintenance of a master catalog of courses now kept in notebooks is very time-consuming and expensive. Copying of this material for bulletin publications is also costly. Selective listing as needed for bulletin publication will reduce copy and clerical expenses.

18. Less Clerical Effort. Class schedule copy for a particular semester is now prepared from the class schedule file for the corresponding semester one year previous. This copy must be manually corrected for changes in course offerings which occured during the year. Since the course file in the system will reflect all such changes, this manual update of copy will be eliminated.

    Computer preparation of the State Regents report on courses will eliminate much clerical effort.

    Computer retrieval of the course records for updating following State Regents action will save much of the time and effort now spent in updating our course notebooks.

    Computer listings of courses for bulletin copy will eliminate the time-consuming copying of our master course notebook.

19. Avoidance of Redundancy. Presently, course records are maintained in both Academic Bulletins and Classroom Scheduling. Both of these offices will be able to use the same file records under the system.

20. Improved Access. All terminals will have access to the latest information on all course offerings.

Application: F - Academic Resources - Instructors

Definition: A file of instructors included in any of the Class Schedule offerings is maintained with the capability of entering appointment, assignment, and activity data. Other faculty may be added for faculty activity analysis or other personnel reporting. Records of office locations and number of advisee assignments by undergraduate and graduate levels are maintained.

Potential Benefits:

2. Improved Faculty Services. Retrieval of faculty assignment data on classes and advisers will reduce the need to request faculty to report this information to the deans' and Provosts' offices.

3. Improved Staff Services. An up-to-date file of instructors with appointment, class and adviser assignments, office location, and activity data will assist the Provost's Office in faculty activity analysis. It will also assist deans' offices in adviser assignments.

14. Improved Utilization. Accurate and easily retrievable data on adviser assignments will assist in more equitable distribution of these assignments over the faculty and will permit better utilization of faculty for this important service.

22. Improved Accuracy. Relying on computer records for faculty assignment data rather than faculty self-report will improve the accuracy of the data used in faculty activity analysis.

Application: G - Academic Resources - Class Schedule

Definition: As a new term approaches, the class schedule files of the corresponding previous term are matched against the master course record and a tentative class schedule record is generated. A listing of the tentative class schedule for the new term is prepared and sent to departments. Changes in the tentative schedule returned by the departments are entered via CRT's.

As departmental requests for rooms are entered, screens displaying earlier assignments are accessed to determine availability. Any number of meeting times, meeting places, or instructors may be coded for any class. The Final Class Schedule is listed for reproduction.

A record of each student assigned to a class is maintained for immediate retrieval at a CRT. Details of all data associated with a particular class may also be displayed.

Potential Benefits:

3. Improved Staff Services. The need for a shortened turnaround time for the preparation of the Class Schedule has been articulated often. The system will provide the capability of significantly shortening this time required for the entering of the data into the computer which has been a major factor in the scheduling of the call for class schedule copy.

5. Improved Schedule and Course Offerings. We presently attempt to stay close to the traditional semester schedule because of the difficulty in processing other variations. It will be easier for departments to make desirable changes in course offerings and even to try various alternative master schedules in order to work out better schedules and course offerings.

6. Improved Planning. The system provides a method to identify the department's teaching responsibility for cross-listed courses. We currently have no link to identify these cross-listed courses and therefore lack the ability to give the departments credit for all the teaching areas for which they may be responsible.

7. Improved Reporting. Greater control over class time and size, faculty assignment, and classroom availability will insure the correct reporting of our instructional load data to the State Regents Office. It will also provide new reports on courses, instructional load, and classroom utilization, which will be of greater value in improving the management of our facilities and instructional resources.

10. Improved Control. Immediate access to accurate data on classroom assignments will improve control over the assignment of this improtant resource.

Application: G - Academic Resources - Class Schedule

Benefits: (cont'd)

14. Improved Utilization. Better scheduling results in more complete use of classrooms, faculty and other instructional facilities. As classrooms are better used, some may be turned over for other use. Scheduling of irregular terms will also increase facility utilization.

18. Less Clerical Effort. Entering departmental requests for time and room as the first step in the process will eliminate much clerical effort and sorting in the preparation of a class schedule. Immediate verification of room availability and instructor ID will avoid much clerical effort in correcting these problems later. Departmental requests will go directly to Classroom Scheduling and may be accessed by deans via terminals shortly thereafter, eliminating a stop in the cycle.

19. Avoidance of Redundant Files. Deans need not keep their own records on class scheduling.

21. Expanded Data Base. The Class Schedule File on our computer is currently not adequate for our needs. It does not provide for an interface with a master catalog of courses, or a master file of facilities, or a master file of instructors. These deficiencies will force us to reprogram this system unless an alternative method is provided. The system will eliminate the problem.

24. Improved Speed of Response. Late changes in schedule may be accommodated easily. Scheduling irregular terms during the year will also be more convenient.

25. Timely and Current Data. Each semester the time required for collection of the data for the term class schedule is very long, a problem of considerable concern to most department heads. It will be possible to directly key this information to the computer and provide classroom charts and other evaluation data for our use. Consequently, the lead time required for the preparation of the class schedule will be reduced substantially. These procedural improvements will relieve our overpressed staff and would at the same time allow them to spend more time with important decisions on classroom utilization and the scheduling of classes throughout the day. Eventually, the class schedule data could be made available to the departments by terminals centrally located in Dale Hall, Physical Science, and other key places on campus. Allowing departments to inquire as to the status of their particular course on a day-to-day basis would be an extremely valuable procedural improvement.

Application: H - Registration, Advising, and Schedule Changes

Definition: Registration materials are produced and distributed for an "advance" registration period. Deans and departmental offices are provided with admission information or previous student records data either through terminals, hard copy print-outs, or microfiche.

After advisement, students turn in their requests for schedules. Tabulation of student requests is made and distributed and class offerings are adjusted to optimize satisfaction of student requests. Students are then "sectioned" into classes and the resulting schedules are mailed. Students are given the opportunity to change these schedules by coming in, telephoning, or writing. These changes in schedules are entered via terminals throughout the interview period between advance registration and the "regular" registration period just prior to classes beginning.

During regular registration students are given class cards in an "arena" operation. Class cards are loaded to the files and class rolls are produced.

Adds and drops after classes begin are entered via terminals. Terminal entry of a requested class initiates edit checks for class availability, open sections, and schedule conflicts. If an edit diagnostic is encountered, the screen is returned with the error indicated. Schedule conflict edits may be overridden and retransmitted. Entry to a closed section is not permitted until the section is opened by increasing availability through another screen.

Acceptance of course registration is indicated on terminals by a return response showing the entire schedule, instructor and room assignments, and total hours registered. The transaction is "exploded" to update all appropriate files including the counters for the class.

Through the use of the "incremental close rate" data element, classes may be closed on an incremental basis.

Potential Benefits:

1. Improved Student Services. The system will provide a significantly improved capability to give the student the schedule of courses he seeks. It will reduce the time he must spend in registration, and it will give him better records of his registration to avoid problems such as schedule conflict, finding his class at the right time and place, conflict in scheduling of work such as practice times, attending the correct section, provision for lunch, and confirmation of his registration status (Pass/Fail, auditor, etc.) in a course.

2. Improved Faculty Services. It is expected that faculty involvement in our "arena" registration and in the add/drop process after classes begin will be materially reduced. Improved records on student past performance will assist in advisement.

Application: H - Registration, Advising, and Schedule Changes

Benefits: (Cont'd)

3. Improved Staff Services. Deans' offices will be provided more and better data with better access to all data to assist in their responsibilities in advising and counseling of students.

4. Improve Instruction. Better control over class size will enhance the teaching environment.

5. Improved Schedules and Course Offerings. It will be more convenient and possible to make schedule changes. This will result in getting a student out of a class for which he is not suited, or which is not what he anticipated, and into a more suitable one.

6. Improved Planning. Tabulation of student requests for courses will result in better utilization of staff and faculty resources. Improved balance of section size will enhance teaching environment.

7. Improved Reporting. Reports to deans' and departmental offices will be improved in two ways; first, in timeliness, and, secondly, in content. Additionally, new data such as admission data and past enrollment and performance data will be made available to advisers to assist in the advisement proces.

10. Improved Control. The system will provide better control on class size, avoidance of scheduling of conflicts, balancing of section size, oversubscription of classes, adjustment of class size and class rates as registration proceeds, numbers of students who get "desirable" schedules, and control on enrollments in cross-listed courses.

11. Improved Monitoring. Since it will be possible to inquire as to a section or course enrollment at any time, it will be possible to monitor these enrollments throughout the registration process. This will improve opportunities to adjust plans as the needs of colleges and departments change or in response to changes in student demands.

12. Orderly Growth. Registration loads can increase to incorporate increased enrollment, registration in programs such as Advance Programs, Correspondence Study, Medical student, and off-campus Liberal Studies without a corresponding increase in staff and clerical effort to do the job.

17. Reduced Costs. Reducing faculty involvement in the registration process will basically reduce the cost of registration to the University although the savings will accrue to other budgets.

18. Less Clerical Effort. Much staff time is now required to complete registration processes. Much of this effort will be eliminated.

Application: H - Registration, Advising, and Schedule Changes

Benefits: (cont'd)

19. Avoidance of Redundancy. Files of registration information may be shared with other offices on the system, thereby avoiding duplicated files.

20. Paper Reduction. Deans' offices files on student registrations will be unnecessary since the current data will be available at the terminal.

22. Improved Accuracy. "Echo" checking of add/drop data entered at the terminal will materially improve the accuracy of the data entered. Edit diagnostic will avoid errors of data entry.

23. Improved Access. Access to current data on registration will be extended to deans and other offices.

24. Improved Speed of Response. Results of add/drop transactions will speed the recording of this data on student billings, class rolls, and enrollment reports.

25. Timely and Current Data. Last minute changes in the schedule can be accommodated easier than with present techniques. An increase or decrease in instructional staff, for example, can be handled since immediate recall of all registrants (with addresses) in a class can be made.

Application: J - Student Billing and Check-In

Definition: After classes begin, a schedule and fee invoice are prepared along with a form for edit by exception updating the student's record. These materials are distributed in an arena operation during the second week of classes. The student completes the edit document, turns it in, and pays full fees. Identification cards are validated or pictures are taken for new ID's.

Potential Benefits:

17. Reduced Costs. Presently, we are producing a separate Schedule and Fee Invoice form and a Student Information Card for student editing of our data base. A combined form will reduce forms cost, computer printing time, forms bursting costs, and the manual collating of documents.

18. Less Clerical Effort. The manual collating of the Schedule and Fee Invoice, which now requires 80 hours of clerical effort per semester, will be eliminated.

22. Improved Accuracy. Frequent review of data provided to us is necessary if the data is to remain current. Student editing of this data at the time of check-in will provide this updating.

Application: K - Student Records

Definition: As admitted applicants register, their records are transferred to the "student" file. Student schedules, fee invoices, edit by exception documents, statistical reports, class rolls, advisory assignment lists, grade lists, grade reports, and advisory transcripts are produced from these records. Additionally, these records are available for on-line inquiry by offices having access to terminals.

If a student leaves the university, is suspended, or graduates, his record remains on file for three years. After an absence of three years, the student's records are moved to tape and are included in a microfiche product used for security and reference purposes.

Up to nine addresses, including telephone, may be maintained for any applicant or student. Each address is numbered and address flags are used to record the address number to be used for the student's local, home or spouse address, parent's or guardian's address, emergency mail or billing address, address for mailing grades, and two other miscellaneous addresses. With eight of the addresses we may associate a name. The name could be that of the spouse, parent, guardian, or person or agency to be billed. This name field is also used to store maiden and other former names of the student. All former names are cross referenced in the name-locator files.

The address flags indicate the relationship between the address and various functions. The program edits for a logical relationship of all flags and addresses and will not accept data which fails edit tests. Addresses may be copied into other address fields and the student's name may be added to the name field of an address record.
A screen is available to change an individual's social security number. All files are automatically updated and special logs of all number changes are produced for distribution to various offices.

Potential Benefits:

1. Improved Student Services. We have received complaints that students do not know where to report an address change. If they report these changes to the Student Development Office or to a college office, our office does not generally get this information. If they report it to our office, we make an attempt, via paper transactions, to notify all of the offices that might be involved. New data would be available to all offices with terminals as we update. In this way, we would have immediate distribution of address changes without requiring our students to go from office to office reporting this information.

3. Improved Staff Services. Up-to-date address data, available in machine processable form and coded as to intended use (writing, billing, emergency, parental contact, etc.), will materially assist University offices in contacting students.

   Adequate cross reference records on maiden and former names will improve access to records. Adequate notification of name and student number changes will assist all offices in maintenance of student records.

   Advisory assignment data lists will assist in assigning new students to advisers.

Application: K - Student Records

Benefits: (cont'd)

6. Improved Planning. More accurate and timely reporting and the flexibility to rearrange statistical data for presentation to deans' and other offices provides administrators better information about the students from which to make short-term decisions and to do long-range planning.

   The availabilty of the student's cumulative record will make it possible to determine that academic policies are being met. It will also verify that applicable academic policies are being implemented uniformly across the various campus departments, schools, or colleges.

7. Improved Reporting. The extensive capability to record address data will improve our ability to provide adequate addresses for mailings and other contacts.

8. Orderly Growth. Presently, we are considerably constrained in the expansion of our Student Master File, making it difficult to add new students, such as those registered through the Advanced Program format. With the system, the only constraint will be the availability of adequate computer hardware, i.e., as our file expands, the files can also expand. Presently, this is not possible without a considerable amount of reprogramming. In addition, the system will allow us to begin adding new categories of students to our files, eliminating the need for separate records systems as proposed in the Advanced Programs area.

17. Reduced Costs. Supporting the student records function of the many college offices and other offices on campus is an expensive proposition. Generally speaking, each college office has created its own student records system. Their need for this data for advising and for degree check is real. However, given a terminal, these offices could eliminate much of their records-processing cost, since all of the information would be immediately available to them without hard-copy records. Additionally, they would have available to them, almost immediately, any changes in the record that may occur.

    It should be noted that the Office of Student Development, the Housing Office, and other related offices also have considerable need for student information and require their own records. Again, access to our Student Master File on the computer will considerably increase their efficiency and reduce their record costs.

18. Less Clerical Effort. Records on new admissions are forwarded to colleges as they are admitted. After registration, most colleges purge these records of "no shows". Automatic "purging" of these records on computer files will obviate the need for most of this effort.

    Once the student record system is installed, the staff needed to process and maintain student records can be maintained at current levels despite increses in enrollment and in the numbers of student records maintained.

Application: K - Student Records

Benefits: (cont'd)

19. Avoidance of Redundant Files. The University has made the mistake of proliferating its student record system on campus by not providing adequately for a central records system. Clear manifestations of this development were the organization of the Health Sciences Center Admissions and Records Office in the early 1970's. Fortunately, the University has decided to integrate these separate records systems and good progress is being made to this end. However, the problem of building a central records system accessible to each of these three offices is substantial. Terminals will eliminate the difficulties almost entirely.

    Adequate cross reference records and notification of student name and number changes will prevent creation of second redundant files.

21. Expanded Data Base. The system will permit the computer coding of those students who are involved in various veteran programs. At present, such coding is not available even though over 1,900 students are now being certified to the Veteran's Administration. Other Financial Aids record keeping for other programs will also be improved considerably.

    The present student data base on the computer is not adequate for the needs of the Graduate College and other departments. In order for us to modify our present data base to integrate this data, it will be necessary for us to redesign and reprogram much of the system. The new data base will provide for the accumulation of most of this new information. New reports and improved facility in handling the large amount of data involved should materially assist our graduate admissions and records effort.

    Expansion of data elements is provided in the system. University reporting performance will be enchanced since data elements will encompass NCHEMS, AACRAO, and other standards. Our present computer system does not provide for the accumulation of records of course work. Consequently, each semester this information is listed on pressure sensitive labels, affixed to hard-copy records in our office, and purged from our computer files shortly after the end of the semester. The new system will ultimately provide for the accumlation of this data on the computer and the production of official and advisory transcripts on an as-needed basis. Gradually, as our data base expands, we will replace our hard-copy records and move completely to our computer files.

22. Improved Accuracy of Data. The system permits improvement in file update capability for specific areas such as class, college, major, cumulative credits and GPA, semester credits and GPA, probation and dismissal status, and other important information on the Student Master File.

23. Improved Access. Most data records on students will be available to deans' offices via terminals. Much of this data is not presently made available to them or does not exist.

25. Improved Timeliness and Currency. Many changes in the data base are made to the files immediately upon notification and are subsequently available at other terminals or in system reporting.

Application: L - Grade Reporting

Definition: As the end of the term draws near, grade lists are prepared and distributed to the departments. As the lists are returned with grades, they are entered into the system. Entry of grades initiates edit checks for accuracy, validity and completeness; i.e., classes may have differential grading patterns such as satisfactory/unsatisfactory grades for some and not others; student pass/fail options are observed, etc.

Several times during the grade reporting cycle the system queries each registered student's record to determine if all grades are in. If so, a grade report is produced and the record is tagged accordingly. If a request for a transcript with grades has been entered, these request are also fulfilled.

After all grades are in, lists of delinquent grades are produced. Also, retention rosters are produced and distributed to deans' offices. Probation and suspension decisions are entered by the deans through their terminal. It is possible that probation and suspension notices can be computer produced and mailed by the dean's office. Statistical grade summaries and analyses are prepared and distributed.

Potential Benefits:

1. Improved Student Services. A grade report will be produced and mailed as soon as all grades for a student are in; i.e., without waiting for the end of the grade reporting period to run all grade reports. Similarly, the production of official transcripts to be sent to other institutions and agencies will be expedited.

3. Improved Staff Services. Statistical analysis of grades will obviate the need for deans or departmental chairmen to conduct their own analyses.

   Grade information can be made available to counselors in advance of a counseling session. In addition, exception reports showing weak students, or those with unusual grade patterns, can provide information from which counselors can initiate special sessions.

5. Improved Schedules and Course Offerings. Curriculum consultants need to know how students are performing in the various courses in order to improve the design of those courses.

6. Improved Planning. Earlier notification of grades will assist students in planning for the next semester's activities.

9. Improved Report Scheduling. At the end of a semester we accummulate a large number of requests for transcripts to be sent after grades are recorded. The system will allow faster production of transcripts.

10. Improved Control. Grade analysis information may be used to insure that institutional marking policies are being adhered to and that they are applied uniformly accross the various units within the University. The grade reporting information can be used to provide various repiots and analyses of grade distribution based on program level, student level, course, teacher, and sex.

Application: L - Grade Reporting

Benefits: (cont'd)

11. Improved Monitoring. Monitoring of files for late grades will be improved. Lists of grades not in after the deadline will be automatically produced.

18. Less Clerical Effort. The automatic, accurate, and timely preparation of grade reports, transcripts, and grade analyses can substantially reduce the clerical effort now involved in producing such reports.

Application: M - Advisory and Official Transcripts

Definition: The system will produce either an advisory or official transcript. Advisory transcripts include admission and transfer evaluation data of value to advisors in addition to the OU course work record. Official transcripts include OU course work research and summaries of transfer work. Requests for transcripts may be addressed for insertion in window envelopes.

As requests for transcripts are received, they are entered via the name locator screen. Requests for special grade reports or copies of student class schedules are also entered via this screen. A "reports requested" screen may be accessed to determine status of all such requests prior to actual production.

Potential Benefits:

1. Improved Student Services. Improvement in response to requests for transcripts, special grade reports, and copies of lost schedules will assist students in many ways. A complete advisory transcript will assist them in their planning of degree programs and term schedule requests.
2. Improved Faculty Services. Advisory transcripts will make it possible for faculty to have needed information readily available for advising.
3. Improved Staff Services. Several years ago, because of costs, we were forced to discontinue providing advisory transcripts to colleges on a routine basis. The system will permit the distribution of such transcripts at low cost. Since deans' offices will have access to the data via terminals, it is probable that such transcripts may be directed to a student's adviser--moving the data to that important decision point.

Application: N - Degree Planning

Definition: Student degree plans may be entered and maintained by college or departmental offices. An initial plan may be established by copying degree plan "models" or "standards" stored in the system. Each course the student has taken or is registered for, OU, or tranfer, is entered into the degree plan record with coding indicating its applicability to the degree plan and whether or not the course has been completed. Establishment, security and maintenance of the record is delegated to dean or departmental offices. The college may specify that the existence of a degree plan for a student is mandatory.

Potential Benefits:

1. Improved Student Services. Monitoring of the student's degree plan will help to insure completion of all degree requirements in the normal time required for the degree. It will also assist the student in his selection of a schedule for a particular term.

2. Improved Faculty Services. Computer monitoring of degree plans will preclude the necessity of faculty monitoring records of such planning.

3. Improved Staff Services. Availability of degree plans from terminals will enhance deans' office maintenance of such records.

6. Improved Planning. Continuous availibility of the current status of each student degree program will greatly improve student counseling and advising. Course deficiencies in completing degree requirements may be easily detected at an early date so that necessary changes in planning may be made. Edits may also be made to restrict a student from registering in courses not contained in his program. This capability also facilitates planning of course offering requirements of future term in relationship to student needs.

11. Improved Monitoring. Monitoring student records to determine progress toward degree goals will be facilitated.

12. Localize Responsibility. Existence of degree plan records and decisions in readily accessible form will allow deans' offices to assume greater responsibility for ascertaining that degree requirements have been fulfilled.

16. Extend System Familiarization. Existence of coded degree plans will insure that degree requirements are clear and will improve communication of these requirements to students.

21. Expanded Data Base. All data and capability involved in this application is new and provides a significant new service from Admissions and Records.

Application: P - Statistics and Institutional Research

Definition: Statistical data on current headcount and credit hour enrollment are instantly available through the terminal. This data is stored in a file designed to provide this data. Distributions by sex, student level, college, major or program, program type, program option, residency, and racial origin are available. College and program major, type of option are uniquely related to the HEGIS taxonomy and degree codes. As students register or withdraw, these files are automatically updated.

The report generator of the system provides the user with the cabability of selecting and formating reports containing any of the data elements in the system. The reports are produced on lists, labels, cards, tape, terminal, or microfiche. Report requests are stacked in the batch job stream at the Computer Center and are run as soon as time is available.

College, major, program type and program option codes are related to degree codes. All of these are related to HEGIS taxonomy codes for reporting to state and federal agencies.

Potential Benefits:

3. Improved Staff Services. Adequate statistical data will assist many offices in the answering of the many questions and questionnaires they receive every day. The report generator will also assist the staff in selecting and arranging data of value in decision-making in their area of responsibility.

4. Improved Instruction. Adequate data on teaching loads, student loads, student background and abilities, and student performance will assist faculty in their instructional responsibilities.

6. Improved Planning. The report generator will provide University administrators with a direct and efficient interface with the computer data files in order to secure the information required for planning.

7. Improved Reporting. There is general agreement that data on students, facilities, courses, and instructors is now inadequate for the University's needs. The system will provide a substantial improvement in the routine reporting of such data.

   Adequate coding of students as to college, major, program type and program option will improve our reporting of program data to the State Regents and to HEW. (including college comparison studies)

8. Facilitate Special Reports. The system will provide the capability of generating data designed to answer specific questions or concerns on an as-needed basis and in a short period of time.

20. Paper Reduction. Availability of data via terminals and by special reports rather than routine reports will reduce the need for many listings now produced to capture and distribute the data.

Application: P - Statistics and Instructional Research

Benefits: (cont'd)

21. Expanded Data Base. We are currently redesigning our present system and writing new programs to more adequately integrate the large amount of data provided by the ACT assessment program. We will be only partially successful. The availability of all of this information in computer-accessible form will provide new and much needed information to our advisers and counselors. The data will also be a valuable source of information for academic and institutional research. The system will integrate much of these data onto on-line files or make possible efficient computer reference to magnetic tape files with the remaining data.

22. Improved Accuracy. The general improvement in accuracy of data throughout all applications on the system will also result in improvement in the accuracy of statistical reports. In addition, it is expected that the design of tabulations will be improved substantially to permit cross footing to known totals.

23. Improved Access. Access to statistical data on admissions and enrollment will be extended to deans' and other offices. Additionally, the report generator capability will allow these offices to design and run their own statistical reports.

24. Improved Speed of Response. Admission and enrollment totals will be available for retrieval at terminals at all times.

Application: Q - Integration of Student Records

Definition: Student records for the three "campuses" (Norman, Continuing Education, Health Sciences Center) are maintained on the system. The Admissions and Records Office of each campus (OAR - N, OAR - CE, OAR - HS) may inquire into the records of other offices under security control but may not update other office records. Interoffice communications are facilitated by interterminal message and broadcast capability. Continuing education units (CEU's) and clock hours credits are incorporated to supplement regular credit hour registration and statistics.

Batch reports are defined uniquely for each campus. Batch reports for the Health Sciences Center are produced on a remote terminal in Oklahoma City. Multiple campus registrations are possible.

Term codes identify the start and ending dates for each class offered. Batch reports may be produced as required based on the term code.

Potential Benefits:

1. Improved Student Services. Student registration in courses at more than one campus of the University system will be facilitated. Errors frequently brought about by multiple campus registrations will be reduced.

3. Improved Staff Services. Computer system capablility for OAR - CE will be totally available. The capability for OAR - HS will be enhanced considerably. Both offices will have records of OAR - N available for reference which will facilitate their efforts.

4. Improved Instruction. Presently our operations are basically geared to the fall, spring, and summer terms. The system will allow us to adjust our operation to the needs of the instructional program, i.e., scheduling of reports may be geared to the start or stop of a particular program whether it be weekly, monthly, annually, or whatever. Improvement of admission, registration, and grade reporting procedures for these irregular terms should assist the instructors and free them of unfortunate paper work burdens.

5. Improved Scheduling and Course Offerings. Flexibility in term scheduling will assist in the scheduling of courses.

7. Improved Reporting. Extension of computer services to OAR - CE will assist them in improving their reporting capability. Enhancement of the computer support to OAR - HS will also assist them in fulfilling their reporting requirements.

   Often a student will establish a record in more than one OAR office. When this happens, he is required to request a transcript from each office. Integration of the records systems will prevent this from occuring.

   The particularized needs of OAR - HS for reporting will be observed providing better reports than presently available from a system basically geared to OAR - N.

Application: Q - Integration of Student Records

Benefits: (cont'd)

10. Improved Control. Better control of registrations in irregularly scheduled courses (short courses, off-campus class, BLS classes, correspondence study, workshop, institutes, etc.) will be possible since the system will adjust to the term dates.

12. Orderly Growth. Without extensive reprogramming and expansion of file capability our present computer sustem cannot handle the addition of any OAR - CE records such as Advanced Programs, Correspondence Study, or off-campus classes, or OAR - HS records such as medical students.

13. Improved Record Security. Presently OAR - CE has no security from the catastrophic loss of their records. Since the system provides such security (See System Security), no special efforts to microfilm or otherwise secure current and future records will be needed.

14. Improved Utilization of Resources. The system extends computer support to all three offices of Admissions and Records utilizing the same computer and the same system. Other approaches usually require separate computer support and/or systems.

15. Localize Responsibility. Each OAR will have responsibility for the maintenance of its own data base and the scheduling of its own reports.

16. Extend System Familiarization. All three OAR's will be using the same system and differences in operational procedures will be minimized, thereby improving understanding and communication between the offices.

17. Reduced Costs. Transporting of admissions, registration, and grading materials to OAR - HS will be substantially reduced.

18. Less Clerical Effort. Time spent in reproducing records for transporting between the three offices will be reduced.

    Computer support for OAR - CE will reduce their clerical requirements.

19. Avoidance of Redundancy. Presently a considerable amount of redundancy in records exists between the three OAR offices. Integration of the files will eliminate the need for such redundancy in the future.

23. Improved Access. Access to records of other campuses will be possible.

24. Improved Speed of Response. Many inquiries as to records of students in other OAR offices will be immediately answered via terminals.

Application: R - System Operation and Control

Definition: The system is completely "real time", i.e., all records are immediately available at terminals, and all edits and logical functions take place at the time of data entry or update.

The system "explodes" the data by generating multiple transactions from a single input transmission. Logical edits are specified by the system and before any data is entered to the file, it must pass the edit requirements. Positive edit response is made to all transactions entered by the terminals. Any error detected will be indicated in the response. All or part of the update on the screen containing the error is repeated by the system.

All system codes are maintained in tables which may be retrieved and changed via terminals.

Batch reports may be requested via screens. The variables (parameters) associated with each batch report are entered by the user as batch reports are requested. Scheduling of reports is by day and under the control of the user.

A single alphabetical name locator file and screen allows the user to find the Social Security number for retrieval of any record on the prospect, applicant, student, or instructor record. The file is accessed by keying any part of the last name and "paging" through the file from that entry point. Paging will generally be through all categories of records, prospects, applicants, or students, or only one or more categories at a time.

A student's record may not be deleted from the system if the student is or has been registered in a course. Prospect and applicant records are "aged" and "purged" if they never register. A course or class may not be deleted if a student(s) is (are) enrolled. Logical relationships of records are verified during all transaction processing.

Log records are prepared for each transaction made. Log records are "dumped" each day for user review. These records provide a complete audit trail of all transactions made to records.

Complete backup and recovery capability exists. If the computer system goes down, restart is possible without loss of data or transactions. If files are inadvertently destroyed or if a program has been inadvertently operating in error, restart at any earlier point is possible with "replay" of all transactions since that point. Magnetic tape library procedures insure that checkpoint restart records and transaction log records are stored at remote sites in the event of the catastrophic loss of the basic computer files.

The system permits the querying and viewing of messages sent between users of the system. A message may also be "broadcast" to all terminals or selected terminals in the system.

Complete line and terminal status data is available at terminals.

Terminals can be "dropped" (placed off line) or polled (placed on-line). Malfunctioning terminals go through "penalty boxes" to lessen the detrimental impact on system response time. Inactive terminals are automatically logged off.

Systems' "aborts" due to system checks, program checks, dumps, etc., do not abort the system for the user. The user is notified of the existence of the problem but may continue to operate.

Application: R - System Operation and Control

Potential Benefits:

1. Improved Student Services. Uncertainty as to a student's status will be eliminated, permitting more responsive action in handling his requests and needs without error.

9. Improved Report Scheduling. Daily scheduling of reports by the office requesting the report will make report scheduling simpler and more responsive to changing needs.

10. Improved Control. The ability to retrieve all records at terminals will permit better control over all aspects of admissions, registration, and records.

    Many operating systems will "abort" programs if system or program checks and/or dumps occur. This system will allow the user to continue processing without interruption.

11. Improved Monitoring. Log records will provide complete monitoring of all transactions. Errors in data entry may be traced to their source.

13. Improved Record Security. Loss of records due to system or operational failures will be prevented by the backup and recovery procedures. In the event of the catastrophic loss of current computer files it will be possible to restore files to within one week of the loss.

    System's control over deleting of students, courses, or classes will insure against the inadvertent loss of data.

14. Improved Utilization. Efficient utilization of lines and terminals is insured by the availability of line and terminal status data. Automatic dropping of terminals due to malfunctioning also insures maximizing of terminal efficiency.

15. Localize Responsibility. An office with principle responsibility for a record or data element entry and maintenance will have complete control over that record or data element.

16. Extend System Familiarization. Several offices will share in the creation and maintenance of the student data base and code structure through the terminals. Such sharing will acquaint all users of the system with the dynamic nature of the data base and the concomitant problems of reporting from it. Message broadcasting will also acquaint users with systems problems and facilitate communication.

18. Less Clerical Effort. Checking, editing and correcting of input transactions is now a cumbersome and time consuming task. Positive edit response at the point of collection and entry of this data will materially reduce the clerical time involved in such operations.

    Retrieval of data from computer files now requires a substantial clerical effort to maintain paper records arraying the data. The accessing of the data through terminals using the name locator file will materially shorten the search time for records and the maintenance of the paper records.

Application: R - System Operation and Control

Benefits: (cont'd)

20. Paper Reduction. Presently we produce the Student Information Status Report, Student Information Sheet, Alpha Course Roster and Student Alphabetical List several times a semester. These paper products are intended to array for various uses the data on our computer files. Having this information available at terminals will substantially reduce the need for this paper.

22. Improved Accuracy. Logical edits against tables of codes with response indicating any errors detected will prevent the entry of invalid data. Audit trails will also make it possible to trace the source of the entry of invalid data for future correction of erroneous input. Finally the checking for logical relationships between records will prevent the entry of invalid records.

23. Improved Access. Selective access to all records of the system is a key feature of the system's operation and control.

24. Improved Speed of Response. Scheduling of batch output by day at the terminals will speed the response to requests for reports.

Application: S - System Documentation

Definition: The system provides for complete user and data processing documentation. A data element dictionary, complete description of screens with instructions on how to use them, and complete data element listings with logical edit criteria are maintained on line through the terminals. Computer listings of batch reports, abbreviations, systems, data tables, data files, records, and programs, etc., are also available.

Instruction on the use of the system is available at each terminal.

As program or procedural changes are made, some documentation is automatically updated. Other documentation is readily available through the terminals for ease in updating.

Potential Benefits:

6. Improved Planning. The existence of good documentation permits better systems planning as changes occur in operating needs and demands.

10. Improved Control. Documentation is the key to effective control over the operation of any system, especially computer based systems.

15. Localize Responsibility. Both user and data processing documentation is maintained by the system, forcing a merger of the efforts of both parties. Since the user will be able to monitor the documentation of his applications, he will be able to determine that the documentation is maintained and hence may be held accountable for the documentation of his system.

16. Extend System Familiarization. The existence on-line of an adequate data element dictionary, descriptions and instructions on screens, the code structure (tables), and edit criteria will extend to all users of the system the tools they need to be able to fully understand the system.

18. Less Clerical Effort. The principle difficulty with the maintenance of adequate documentation of a system is the time-consuming clerical effort involved. This system will automatically update much of the documentation and will facilitate the maintenance of the rest without considerable clerical effort.

19. Avoidance of Redundancy. Merging both user and data processing documentation on-line will eliminate redundancy associated with separate documentation systems.

20. Paper Reduction. Since much of the documentation will be on-line, it will eliminate the need for producing many copies of the information for the various users of the system.

23. Improved Access. Access to the system's documentation will be extended to all users of the student data base.

Application: T - System Programming and Maintenance

Definition: Programming diagnostics are available for systems maintenance, programming, and monitoring. Complete operational statistics on files and programs are maintained by the system for analysis and monitoring.

Aborts due to program checks, dumps, etc., do not take the system down. The monitor intercepts "refreshes" the system and indicates dumps available to the programmers (complete dump, hex/characters) via terminals. User changes do not actually change any real files if any dumps or purges are effected in any part of the processing.

Simultaneous to the operational use of the system, programmers can operate from terminals in a "test mode". Programmers may use "test" (substitute) versions of ongoing programs while users still operate with earlier versions. In test mode files are not actually changed, but changes are simulated for programmer convenience in "debugging". Programmers may make temporary modifications ("patches") to programs via CRT for test or maintenance purposes.

The system is completely modular, permitting the addition of new files, applications, or systems. Additions of new applications or complete new systems as facilitated by the systems programming support, file handling capability, and telecommunication facility of the monitor.

Various utility programs available with the systems contribute significantly to the effectiveness of programmers and to accomodations of standards requirements. A data element utility provides extensive manipulation and reportage capability of the system data structure for both the users and the programmers, so that both are interfaced with current consistent definition and terminology. It also services programming with the capability to automatically generate COBOL record descriptions from the data element records. A COBOL source statement library utility provides programmers with many useful tools. Standard data and logic modules can be uniquely maintained and "included" or "copied", (temporarily or permanently inserted), into programs. Many versions, including historical and test, of programs may be inexpensively and simply maintained and documented. Programmers have virtually unlimited capability to sefficiently "copy" any portion of other programs or modules to assist in preparations or modifications of programs, and can automatically perform tasks such as "sifting" a program to locate and change all references to given names, etc.

Potential Benefits:

3. Improved Staff Services. The system extends support to all programmers working on the system by providing them with several new capabilities and "tools" which allow them to perform more productively. It also provides a means for other applications such as payroll, faculty and staff personnel records, library, accounting, budgeting, etc., to be put on-line and integrated with the student information system. The development of these new applications in other areas will be facilitated by the programming support, file handling, and telecommunication facilities of the system monitor. As other applications are added, efficiency is enhanced due to the use of consistent definitions and terms, which increase correct communications.

10. Improved Control. Programming and documentation standards which enforced, insuring consistency and understanding.

Application: T - System Programming and Maintenance

Benefits: (cont'd)

11. Improved Monitoring. Ongoing systems statistics are accumulated during processing. Review of these statistics allows in-depth analaysis of program performance and resource impact.

12. Orderly Growth. The programming of new reports and new applications will extend the system's capability to all offices, faculty, and staff as needs for more and better information grow.

14. Improved Utilization of Resources. The efforts of the programmers on the system will be facilitated since they will be able to work on-line on new programs and/or maintenance of old programs throughout the day. Our systems and programming staff are one of our most important resources for extending the capability of the computer in many vital areas of institutional management.

18. Less Clerical Effort. The copy and sift features of the system will eliminate much keypunching and other clerical effort. Since on-line "debugging" capability is present, many runs for compilations of programs during test or repair phases will be eliminated.

20. Paper Reduction. On-line debugging will eliminate the need for many listings of programs. Much documentation (directions and explanations of screen usage, tables, etc.) will be available on-line. Updated documentation and tables can be added through the terminal.

Application: U - System Security

Definition: Security control consists of two passwords assigned to each user. The first password is system defined, and determines what screen or files are available to that user and the functions (add, change, delete, inquire, etc.) that may be performed by that user with each screen.

The second password is user defined and can be changed by the user at will. This allows the user to alter his security key if he thinks it has been compromised.

Each time a user makes a transaction the monitor supplies the servicing programs with various data associated with the user's security specification. The data provides various organizational identifications and "clearance" indicators, allowing virtually unlimited security specifications.

Passwords are entered at the time of logging a terminal onto the system. Passwords are invisible to an observer. Terminals are automatically logged off after specified time intervals.

Records are maintained for three offices of Admissions and Records (Norman, Continuing Education, Health Sciences Center). Integrity of the files of each office is controlled by the system automatically. Password security is unique to each office. Student records for other institutions may be kept on the system with complete security and integrity of records.

Potential Benefits:

10. Improved Control. Control over systems operations not available before is extended to the Offices of Admissions and Records at the Health Sciences Center and at OCCE.

13. Improved Record Security. Access to specific items of data is controlled by the user. Since he interfaces directly with the data base via the terminals, he may restrict access to that data by any intermediaries such as delivery personnel and keypunchers. Additionally, student records are recognized as a vulnerable target by those intent on disruption. Accidental distruction of student records can also occur. An automated system makes back-up files much more feasible and recovery from loss is more extensive and less expensive.

FEATURE ANALYSIS: PAYROLL/PERSONNEL SYSTEMS

L. Carvel Whiting
Director of Systems
University of Utah

SECTION I

INTRODUCTION

It has been said that "if you can't define the problem, you can't solve it." The purpose of this presentation is to define some of the characteristics and features of payroll personnel systems, in the hope that it may be helpful to someone as they attempt to solve their payroll personnel problem.

The approach taken is to first give an overview of a selected group of systems in use. This is based upon a survey of payroll systems at a selected group of institutions of higher education, conducted by the University of Utah. The results of the survey give an indication of some of the characteristics of the systems currently in use, and suggest that the payroll personnel systems currently in use in higher education are, as a general rule, inadequate. It is felt that payroll personnel systems are beginning to receive, and will continue to receive, a great deal more attention than they have in the recent past. This is particularly true as institutions move toward more sophisticated and complete management information systems which require more complete and accurate data in the payroll personnel area.

Understanding that there is a need to improve one's payroll personnel system logically brings one to the question of whether he should "make" or "buy" a new system. Today that decision is not as clear cut as it was a few years ago, due primarily to the emergence of several high quality and relatively low cost packages available in the market place. As reported recently, ". . .sales of standard packages are increasing at the rate of about 30 percent a year and will total an estimated $175 million in 1974. That's more than one-third of the $500 million total market for 'outside' (externally supplied) software . . .Payroll packages are by far the fastest-selling type of application package to date . . ."(1)

Since more and more institutions are looking at the possibility of obtaining a package to solve their payroll personnel problems, a brief discussion and overview of some items to be considered in obtaining a package are discussed.

Finally, a list of some of the more salient features or characteristics of a payroll personnel system are included.

---

(1) How To Buy Software Packages, October 1973 Supplement to Datapro 70, p. 70E-010-20a., Datapro Research Corporation, Morestown, New Jersey.

SECTION II

OVERVIEW OF SYSTEMS IN USE

UNIVERSITY OF UTAH SURVEY

The overview presented here is a summary of a survey conducted by the University of Utah in 1972 and 1973. Although this survey had a relatively small population of 30 institutions, due to the selection criteria it is felt to be fairly representative of one segment of the higher education community.

A. Background

Perhaps it would be appropriate to discuss briefly the events which lead up to the survey. For the past several years it has been generally conceded at the University of Utah that a new payroll personnel system was needed.

The payroll system currently being used at the University was initially installed about 11 years ago. It was written for use on a card-oriented IBM 1401 computer, and consists of several hundred programs written in four different computer languages with little documentation. In spite of these factors, there is a teleprocessing front-end to the system and interfaces with several other fairly sophisticated computerized systems, including a comprehensive on-line financial system.

In 1971 an attempt to design and build an in-house system was terminated after about one man-year of effort, due to inadequate progress. In a review later that year the following ways of obtaining a new system were suggested: 1) Obtain a system (package) from another university, 2) Do a one-for-one conversion of the existing system to COBOL, or 3) Build a new system in-house.

After reviewing these alternatives a decision was made in 1972 to pursue the first alternative, which was to attempt to obtain a system from another university. Accordingly, a survey was undertaken with the intent of finding a good payroll personnel system which would come close to meeting the needs of the University of Utah.

B. Survey Approach

The survey consisted of the following:

1. Determination of the institutions of higher education to be contacted in the survey.

   The initial list of institutions to be included in the survey was selected from those institutions which had a teaching hospital and which ranked high nationally in the amount of federally-funded research. As the survey progressed a few other institutions were also contacted when it appeared that they might have a system worthy of consideration. It is acknowledged that the survey population was small (30 institutions), and that some institution not included in the survey might possible have had the desired payroll system. However, in justification of the selection criteria, it is felt that the requirement of having a hospital and large amounts of federally-funded research create rather unique problems in a payroll system, and that the survey population chosen was a fair sample of those institutions which share these problems with the University of Utah.

2. Preparation of Survey Criteria

The background and salient characteristics of the University of Utah payroll system were reviewed. In addition, persons involved with the current payroll system and previous design efforts were interviewed. This culminated in a questionnaire which was used as a guide in the survey. However, the actual selection criteria used was broader than, but included, the questionnaire. The systems were screened using the following coarse criteria:

a. Does the system appear to be adequately documented?

b. Do the system design concepts and techniques used appear to be reasonably close to the state of the art?

c. Does the system come reasonably close to meeting the requirements specified in the questionnaire?

3. Conducting the Survey

The survey was conducted in three phases. In 1972, twenty-five institutions were contacted by telephone in a preliminary survey, which was followed by an on-site survey of five institutions. In 1973 an additional five institutions were contacted, most of which were visited on site.

C. Results of the Survey

The results of the survey (shown in Table 2-1) were initially surprising since over three-fourths of the systems were not given serious consideration, due either to an obsolete design or inadequate documentation. Approximately half of the systems reviewed had an old or obsolete design. This apparently reflects the fact that the payroll system was probably one of the first systems automated, and that the emphasis in recent years has been in other application areas. It also suggests that in the next few years

Table 2-1 Results of 1972 and 1973 Survey

Total No. of Institutions
Included In The Survey 30

A. Systems Not Considered	Number	Percent	
1. Old or obsolete design			
a. System included 1401 Emulation	6	20%	
b. Old System (> 9 yrs.)	3	10%	47%
c. Inadequate System	5	17%	
2. Documentation			
a. Wholly inadequate	4	13%	30%
b. Poor	5	17%	
TOTAL	23		77%
B. Systems Available For Technical Evaluation	**Number**	**Percent**	
1. Systems judged as not being compatible with our requirements	4	13%	23%
2. Systems given serious consideration	3	10%	
TOTAL	7		23%
			100%

there is likely to be a great deal of interest in the development of new payroll personnel systems in higher education. The fact that an additional third of the systems had either poor or wholly inadequate documentation is characteristic of the typical in-house developed systems, where documentation is not given very high priority.

Of the seven systems judged to be worthy of further technical evaluation, four systems were quickly judged as not being compatible with the requirements of the University of Utah for various reasons and only three systems were given serious consideration. These three systems were given very careful review by a review team, and one of the systems was chosen for implementation at the University of Utah.

It should be noted that in 1973 we also looked at a few commercial payroll packages being marketed by independent software companies. Generally speaking, we found that the commercial payroll package did not address the kinds of problems faced by a major research institution. For example, these packages had difficulty handling payment for people working on contracts and grants, where they have multiple appointments and are being charged to various account numbers.

SECTION III

SOFTWARE PACKAGES, GENERAL CONSIDERATIONS

This section is not intended to be a comprehensive treatise on the subject of software packages, because there has been much written about this subject (e.g., the Datapro 70 Report referenced earlier, which was used as source material for this section). However, it is felt that a brief discussion of some of the general considerations of software packages is appropriate because it points out some characteristics of the package which are important whether the package is developed in-house or whether it is obtained from an outside source. As was mentioned in the introduction to this report, sales of standard packages are increasing at a very rapid rate and payroll packages are by far the fastest selling type of application package to date. In general, there has been a tremendous upsurge in this area of the software industry in recent years. There are now several sources from which you can obtain software: 1) In-house development, 2) Computer Manufacturers, 3) Users' group (CAUSE), and 4) Software houses, etc.

More and more companies are now giving serious consideration to the question of whether they should "make" or "buy" software packages. There are several things which should be considered as one approaches this question:

1. The time required to complete the project,
2. The total cost and associated risk,
3. The probability of completing the project,
4. The quantity and quality of documentation,
5. The resistance to obtaining packages from the outside.

Before reaching a decision to make or buy a software package it is advisable to consider carefully the relative costs and benefits of the in-house developed software as compared with a package obtained from the outside. This comparison should include both the direct and indirect costs and should include all true over-head costs associated with in-house development. Generally speaking, if there is a software package available which comes close to meeting your requirements, it is likely that it could be installed in a shorter time at less cost and associated risk and with better documentation than if it were done by an in-house group.

It should also be noted that the decision to make or buy need not be absolute. It is possible to buy a package and then tailor it to meet your specific requirements. This is the approach which has been taken at the University of Utah.

When a decision has been made to go through the make or buy analysis, it is necessary to outline the steps to take in performing that analysis. The following procedure is essentially that used at the University of Utah:

1. Determine the requirements you expect the package to meet.
2. Generate a list of packages to be included in the analysis and begin to gather information about their characteristics.
3. Narrow the field by doing a preliminary screening.
4. After narrowing the field considerably, perform a detailed evaluation and comparison of the remaining packages. This should include as detailed an evaluation of all of the features and characteristics as possible.
5. Talk to the users of the package. This is especially valuable if the package ahs been installed at more than one location. If possible, talk not only to the data processing people, but also to the users in the customer departments (e.g., payroll and personnel departments).

After going through the above steps, perform a cost trade-off analysis of the various packages, pick the "best" package, negotiate a sound contract, and install the package.

As noted above, before beginning the evaluation of any software package it is important that you first determine the requirements you expect the package to meet. In addition to the specific features of a payroll personnel system, some of the following factors, which have general applicability, should be carefully evaluated:

1. Does the basic function of the package meet your needs, and how well do the detailed capabilities match your requirements? In other words, does the package have the features you require (see Section IV).
2. Will the package run with your present computer hardware and software? If it won't, then you must consider the cost of either replacing your computer or modifying the package to run in your environment.
3. How flexible (adaptable) is the package? A flexible package may cost more to run, but will probably be easier to install and maintain.
4. Is the package adequately documented? Since the package was not developed by your people, they don't know how it works and it will be very difficult for them to maintain if it is not adequately documented.

5. What is the operational status of the package? Is the package actually in production at the present time? How many sites are now using the package? One of the big advantages of a package can be that it has been installed at several installations and is relatively bug free. If the package is not yet operational, you run two important risks: 1) The package may not be available and ready for use when you need it, and 2) You are likely to be the test site.

6. What is the total cost of acquiring, installing, and using the package? This total cost is the cost that is significant, not just the "purchase price" of the package. In performing this analysis it is important, as mentioned above, that all costs both direct and indirect be included.

SECTION IV

FEATURES OF PAYROLL PERSONNEL SYSTEMS

The term "payroll personnel" implies a system which meets the requirements not only of the payroll department but also the personnel department. To implement such a system many institutions would have to integrate into one system the functions currently being performed by two or more systems. Properly done, this integration of employee systems can result in significant benefits to the institution. It can result in reductions in clerical work load, increased data integrity (less redundancy), a multi-purpose data base, reduced forms processing, and an increase in the efficiency of operations.

In approaching such an integrated system, some institutions have chosen to include the personnel-related functions of the budget department, thereby consolidating three separate functions into one "employment system" and its associated data base which handles all of the information relative to employees at the institution.

Features which should be considered when developing a payroll personnel system are shown in the tables on the following pages. Table 4-1 lists some general features, Table 4-2 lists some payroll features, and Table 4-3 lists some personnel features. Not all of these features are necessary in any given payroll personnel system, but are listed here to stimulate discussion.

Table 4-1 General Features

1. Integration of Management Systems (Payroll, Personnel)

   Reductions in clerical workload
   Increased data integrity (less redundancy)
   Multi-purpose Data Base
   Reduced forms processing
   Operations efficiency

2. Improved Employee Communications

   Fully detailed Check Stub
   Inquiry capability for rapid response to employee questions

3. Multi-employee types and statuses:

   Salaried
   Hourly
   Casual

   Faculty
   Staff
   Students
   Trades

   Full-time/Part-time
   Permanent/Temporary

4. Multi-campus Orientation

5. Payroll, Accounting Integrity

6. WICHE-NCHEMS, HEGIS compatibility

7. Program Budgeting

8. Extensive Documentation (Program and User)

9. Data Element Dictionary

10. Telecommunications Orientation

11. Comprehensive Reporting

12. Report Generation Module

Table 4-2 Payroll Features

1. Time Report/Salary Exception Process by Class of Employee
   - Computer-printed Turnaround documents:
     - Individual Time/Attendance Form
     - Department Time Summary Sheets
     - Clock Cards
2. Time input for Hourly personnel may be on Positive or Exception bases depending upon the Type of Employee
3. Calculations
   - Hourly, Salaried
   - Benefits, Overtime
   - Personal Absence, Sick and Vacation accruals
4. Tax periods (weekly, Biweekly, Semimonthly, Monthly)
5. Variable Payment Schedules (e.g., 9-, 10-, and 12-month payments for Faculty personnel)
6. Cumulative Payroll Distribution by Employee and by Account
7. Voided and Handdrawn Checks and Redistributions processed through the Payroll System to maintain Accounting integrity
8. Earnings Start/Stop Dates by Account or by Appointment
   - Employee's Earnings and Distribution may be dependent upon the Sources of Funds' beginning and ending dates (e.g., for Grants and Contracts) or may be related to the duration of the individual's Appointment
   - Allows the Institution to process future appointments in the current period, minimizing manual paperwork controls
9. State/Non-State Salary Scales
10. Across-the Board Increments processed automatically
11. Stipend and Retiree Payments
12. One-time Changes (Payrates, Deductions, Refunds, Distributions)
    - Permits different values for the current pay period only
    - Requires no maintenance to the Data Base
13. Automatic Distribution and Booking of Earnings and Deductions to the General Ledger through the Payroll Distribution Process
14. Multi-taxing Authorities (Federal, State, City, Municipal)
15. Automatic payments for Retroactive Pay Periods and Vacation Advances
    - Eliminates the requirement for manual calculations
16. Accrual Accounting
17. Unlimited Deduction Types
18. Payroll History for Retroactive Payments
19. Savings Bonds Mechanics and Reporting
20. Automatic Bank Deposits and Reconciliations
21. Extra Compensation Calculations for Life Insurance
22. Automatic Overtime Calculations based on Standard Work Week for the Individual
23. System Parameter Tables to avoid compiled-in Constants and Tax Tables which are subject to change (e.g., FICA %, Hospitalization deductions)

Table 4-3 Personnel Features

1. Affirmative Action Reporting
2. Biographic Information on all Types of Personnel (Faculty, Staff Students)
3. Job Classification Attributes
   - Title
   - Salary Range
   - Benefit Program Eligibility
   - Hourly Rate
   - Skill Category/Academic Rank
4. Retirement Plans/Annuities (includes Tax-sheltered)
   - TIAA/CREF
   - Federal Retirement Plan
   - State Retirement Plan
   - Private Annuity
5. Staff Directory Preparation
6. Sick Leave and Vacation Time Reporting (also Administrative Leave and other Absence hours as required)
7. Annual Employee Fringe Benefits Statements (showing Employee and Employer contributions)
8. Multiple Concurrent Appointments for Salaried Personnel (Dual Appointments)
9. Collective Bargaining Unit Statistics
10. Faculty Activity Analyses
11. Tenure Analyses
12. Termination Statistics
13. Retirement Analyses
14. Benefits Audits for Eligibility and Accuracy
15. Skills Inventory and/or Profiles (by Job Classification or Employee Type)
16. Historical Reporting
    - Multi-year analyses (e.g., Personnel Actions, Payrates)
    - Statistics for Grievance Actions
17. Benefits Reconciliations
    - TIAA/CREF
    - Group Life Insurance
    - Hospitalization
18. Computer-printed Notices
    - Expiration of Term
    - Receipt of Tenure
    - Eligibility for Tenure
    - Benefits Eligibility
    - Salary Increase
    - Deduction Increase
19. Report Generation Module

SEPARATISM REVISTED

Michael M. Roberts
Deputy Director
Stanford Center for Information Processing
Stanford University

SEPARATISM REVISITED

Numerous changes in university computing have occurred since the author's 1971 paper "A Separatist's View of University EDP" was published. A variety of new pressures on both adminis-tative and academic computing centers are forcing old antagonists toward a common future. Included among these are accelerating technology, loss of external support for hardware and software development, and rampant inflation in educational costs of all kinds.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

COMBINED COMPUTING PANEL

Robert C. Heterick
Director
Computer Center
Virginia Polytechnic Institute
and State University

Martin Solomon
Director
Computing Center
University of Kentucky

Gordon R. Sherman
Director
Computer Center
University of Tennessee

Donald J. Wright
Director
Computing Activities
University of Virginia

COMBINED COMPUTING PANEL

Robert C. Heterick
Director
Computer Center
Virginia Polytechnic Institute
and State University

Martin Solomon
Director
Computing Center
University of Kentucky

Gordon R. Sherman
Director
Computer Center
University of Tennessee

Donald J. Wright
Director
Computing Activities
University of Virginia

This panel featured the pro and con arguments for operating the university computing activity from a combined academic/administrative computer operation. Proponents of each point of view were discussed and the factors that lead their campus to the approach they chose. One panelist attempted to assess the impact of future developments in computing technology upon the decision to combine or separate. A serious attempt was made to get to the root of the economic, operational, and political questions that influenced the choice.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the panel members.

TEXT PROCESSING AT VIRGINIA TECH

Ellis P. Bucklen
Associate Director
Computing Center
Virginia Polytechnic Institute
and State University

TEXT-PROCESSING AT VIRGINIA TECH

The Virginia Tech Computing Center has an ongoing publications program, utilizing computerized text-processing techniques, to inform computer users. The scope of work currently encompasses production of reference materials (user's guides), educational materials, proposals for developmental projects, systems documentation, information bulletins, technical papers, and a monthly newsletter.

While serving the internal text-processing needs of the Computing Center, the effort has an additional objective -- that of assisting and informing anyone in the University having an interest in the use of computerized text-processing techniques. This paper discusses some of the problems, procedures, accomplishments, and potential of computerized text-processing, particularly as applied at Virginia Tech.

Initial Text-Processing System

The first computerized text-processing system at Virginia Tech was a home-grown system (Figure 1) in which most of the work was done off-line using punched-cards, subsequently processed in batch mode on the computer. With this system, certain specialized functions of limited scope could be performed. The capabilities of the system were used almost entirely for the production of User's Guides.

In retrospect, the system must be considered to have been more of an expedient distribution system than anything else. Its most noteworthy attribute is that it provided convenient access to useful information. Users having an interest in specific user's guides could obtain copies from the computer for themselves, eliminating the clerical problems and delays associated with other methods of distribution.

While providing convenient storage and facilitating distribution of useful information, the initial system did not provide for ease in

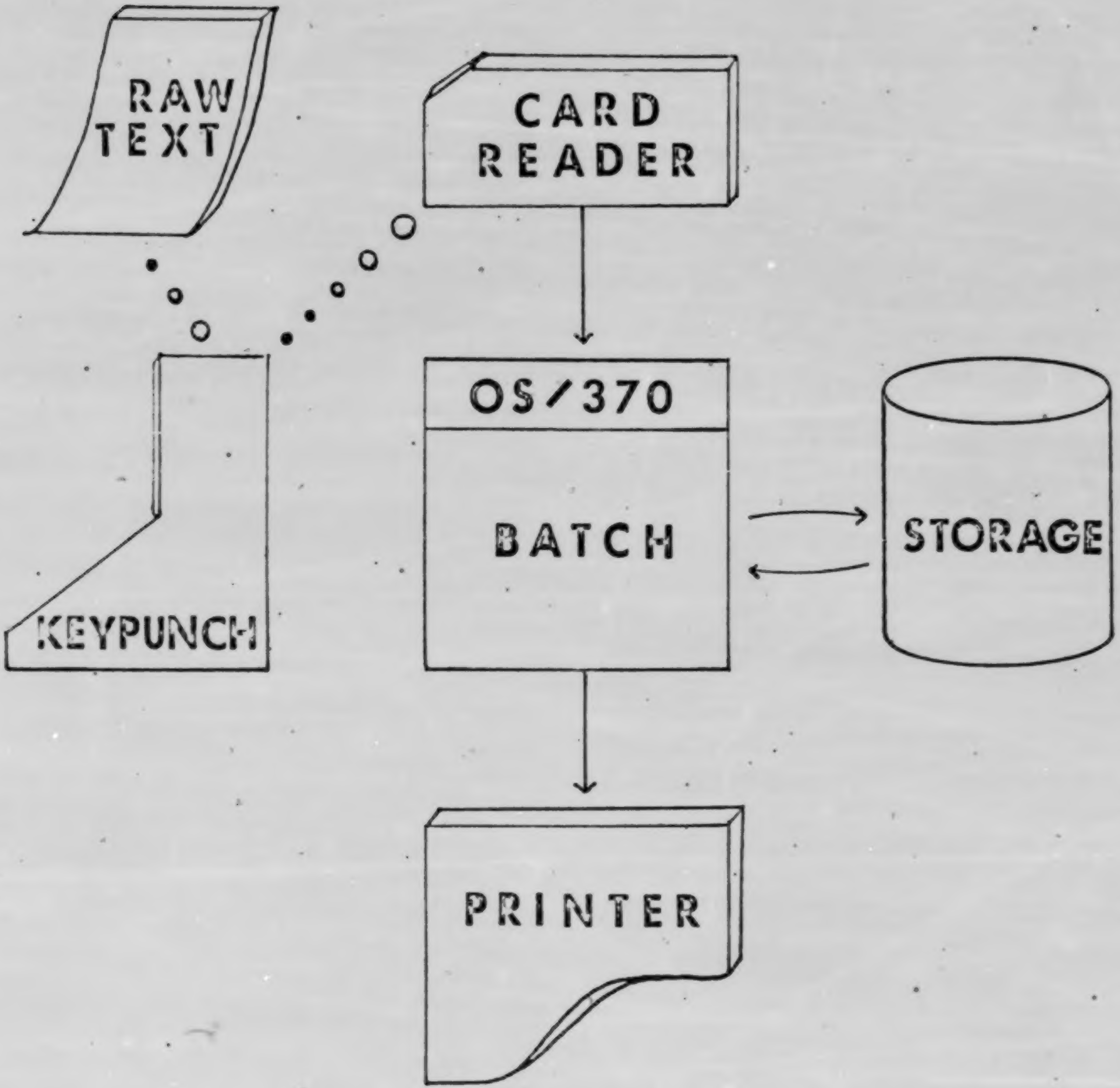

RAW TEXT
CARD READER
KEYPUNCH
OS/370
BATCH
STORAGE
PRINTER

FIGURE 1

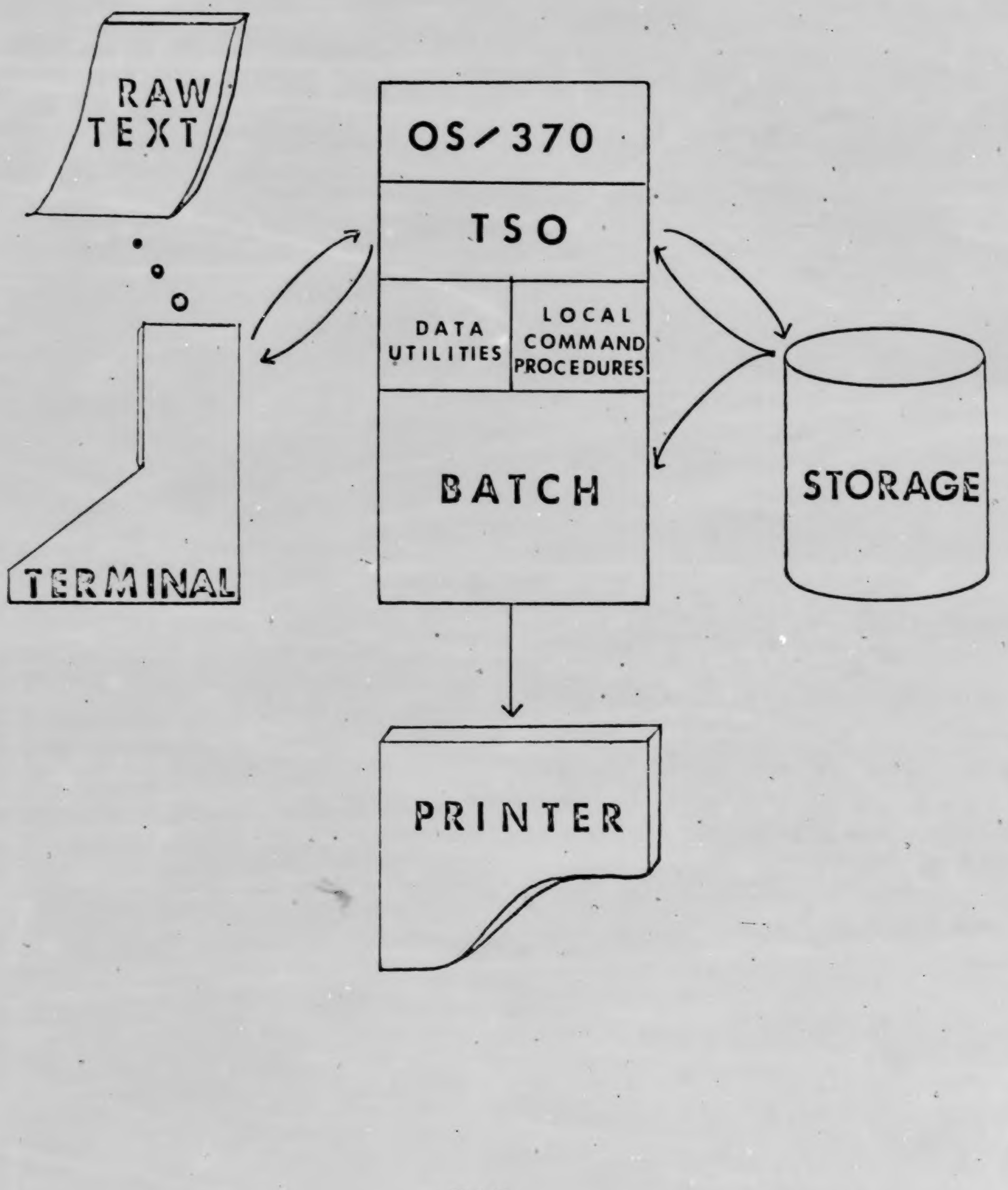
updating stored information. The updating operation was always an inconvenient and time-consuming chore. Consequently, most user's guides were updated infrequently and only rarely were they completely up-to-date.

Current Text-Processing System

The system employed within the last two years (Figure 2) has been an on-line, interactive system, utilizing the Time-Sharing Option (TSO) of IBM's Operating System/370. The TSO Data Utilities program-product is the basic software supporting present text-processing capabilities. These capabilities have been enhanced by locally produced command-procedures. The system, comprised of TSO, the Data Utilities, and local enhancements is referred to locally as the "Text-Editor" and will be so designated throughout this paper.

The new procedures, implemented through interactive access under TSO, provide dramatic enhancements to the Computing Center's document production capabilities. Text can be entered directly into the computer from a terminal and stored on-line in a data set for subsequent manipulations. Formatting commands are interspersed throughout the text to specify the manner in which the text is to be formatted for printing. Upper-and-lower case characters can be used to enhance the appearance and readability of the documents produced. Large-volume output can be directed to a high-speed line-printer to minimize production time. Simple commands are available to change, add, move, or delete text; to rearrange the organization of a document; to justify both left and right margins; and to perform indentations. Through the use of such commands bulk changes can be made to a document without altering any text other than the areas immediately involved, thereby minimizing both the introduction of typographical errors and the subquent proofreading effort necessitated by such changes.

Flexibility and convenience epitomize the capabilities provided by


RAW TEXT
TERMINAL
OS/370
TSO
DATA UTILITIES
LOCAL COMMAND PROCEDURES
BATCH
STORAGE
PRINTER

the Text-Editor in producing documents of various types. To cite an example, the official name of Virginia Tech is the "Virginia Polytechnic Institute and State University," a mouthful, to say the least. Consider for a moment that in certain official publications this name must be used, even though it is unusually long. Consider also what a job this is for a typist in stroking all those keys every time the official name of the University is used. With the Text-Editor, an abbreviation such as "VPI" can be used throughout the body of any publication during the preparatory stages. Forty-eight keystrokes are thereby saved for every occurrence of the official name of the University. When the publication is finished, a single command to the computer will change every "VPI" to "Virginia Polytechnic Institute and State University" (Figure 3). It should be obvious that some people, at least, greatly appreciate this flexibility and convenience.

Updating is now convenient and easily accomplished. A recent reorganization of the Computing Center necessitated a major revision of the "Introduction to the Computing Center User's Guide." Under the old system, this revision probably would have required a minimum of two weeks to accomplish. Under the new procedures, the revision was accomplished in less than six hours, with the expenditure of a fraction of the former effort. This difference is attributable primarily to the manner in which editing and formatting changes are done, manual-effort versus computer-effort respectively.

While the Text-Editor offers impressive capabilities for the processing of textual information, there are still some limitations which are vexingly noticeable. These include the lack of variety in character-fonts and font sizes, restrictions on the use of bullets, and the inability to specify carriage-control on line-printers and hardcopy terminals to effect underscoring. In addition, footnotes are practically impossible to utilize while allowing automatic formatting of text. This can be done only through operator intervention. Also, no control can be

...OFFICIAL NAME OF VT IS THE VPI...

C /VT/VIRGINIA TECH/

...OFFICIAL NAME OF VIRGINIA TECH
IS THE VPI...

C /VPI/VIRGINIA POLYTECHNIC
INSTITUTE AND STATE UNIVERSITY/

...OFFICIAL NAME OF VIRGINIA TECH
IS THE VIRGINIA POLYTECHNIC INSTITUTE
AND STATE UNIVERSITY...

FIGURE 3

exercised on how pages are numbered or on their location on the page. These limitations, as well as the lack of a powerful graphics capability, restrict the manner in which the Text-Editor can be used. However, most of these limitations result from a lack of implementation in the hardware and could be overcome by a commitment to dedicated, more specialized equipment.

Virginia Tech Enhancements to the Text-Editor

To simplify the use of the Text-Editor, a specialized set of command procedures were created locally. For example, the more common commands are performed automatically to eliminate the repetitive entering of sequences of redundant procedures throughout the text-entry process. The step-by-step details of these procedures are transparent to the personnel doing the work.

To facilitate billing for text-entry services, a command procedure was developed to charge directly to the accounts of the individuals for whom work is done. In instances when information for specialized situations is required, messages are issued to prompt operators for the correct response. In order to facilitate problem solutions, a procedure was developed to log software and hardware problems encountered with the system. The amount of time spent on individual projects is automatically logged to enable more accurate billing for work done. Because of occasional system crashes, restart-recovery procedures were developed to prevent the loss of work done up to the point of failure.

A procedure was developed to direct large-volume output to a high-speed line-printer to eliminate the use of slow interactive terminals for such purposes. This procedure was further enhanced by developing the capability to accumulate jobs and submit them in a batch for line-printer output with only one special printer-setup being required. The processing of batched jobs can be invoked either by entry of a command from a terminal or by submission of control-cards through the regular batch job-stream.

Under supplementary software, several enhancements have been provided. One of these is a post-processing program to create a "Table of Contents" by scanning the formatted text of a document. Backup procedures were developed to save text data sets on both magnetic disk and magnetic tape. This provides a measure of security for data sets and also effects transfer of inactive information to less expensive storage media. Other security provisions prevent unauthorized access to data sets. For authorized access, both inadvertent and deliberate destruction are prevented by authorizing access for some in read-only mode while granting read-write access to the person(s) creating the data set.

A program was developed to facilitate the distribution of user's guides stored on the computer. The procedures permit each user to select and obtain the volumes specifically meeting his requirements. The user has the option of getting the requested volumes either in all upper-case or in upper-and-lower case. Additionally, a choice of one of three types of paper may be specified including 8-1/2 x 11 inch paper, prepunched with holes for a three-ring binder.

Other software produces billing information for processing by Virginia Tech's special system-management system. Other software produces management reports on the utilization of both equipment and personnel for individual publications projects as well as for periodic summaries.

Formalization of Publications Services

The Virginia Tech Computing Center is composed of two Divisions, Computer Systems and User Services (Figure 4). Publications support has always been a function of the User Services Division. At first, this effort was a relatively low-priority service performed by a small technical staff, on a catch-as-catch-can basis. User's Guides were

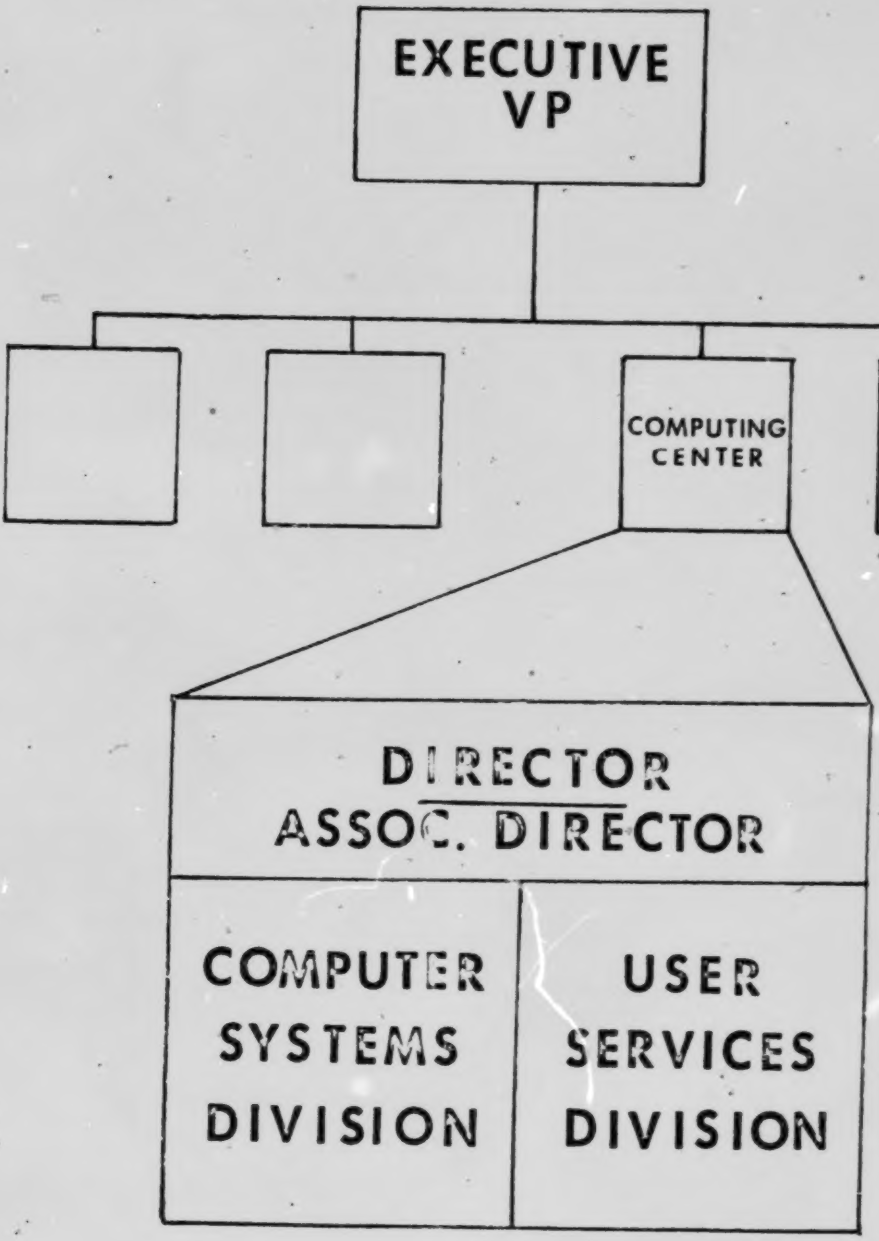

EXECUTIVE VP
COMPUTING CENTER
DIRECTOR
ASSOC. DIRECTOR
COMPUTER SYSTEMS DIVISION
USER SERVICES DIVISION

FIGURE 4

developed as the need was recognized and as the available resources permitted. The responsibility for publication of the monthly newsletter, The Computing Center LOG , also resided with this group.

The publications effort was expanded and formalized about two and a half years ago in the hiring of a full-time "Manager of Publication Services." The task confronted by this manager was a difficult one even though certain procedures had been developed, some of the objectives had been established, and needed improvements had been recognized. Many complicated procedures had to be initiated and numerous technical problems had to be solved while the existing procedures were assaulted by an ever-growing need for additional publications-support.

It was imperative that an interactive text-processing capability be implemented. Thus, software had to be identified and evaluated as did necessary support hardware. Two available text-processing software packages were considered, IBM's ATS (Administrative Terminal System) and their TSO Data Utilities.

The TSO Data Utilities were selected as the software to be implemented for several reasons. First, compatibility with the existing OS-data-set structure, utilized at Virginia Tech, automatically made a large array of general-purpose software available. Second, the capabilities could be implemented under a system that was to extend interactive access to other users of the facility. Finally, the system could be installed without dedicating specific hardware and software resources to text-processing alone. The total resources committed to the support of interactive computing could thereby be minimized while maximizing the availability of text-processing capabilities to the broad spectrum of interactive users.

Expanded Scope of Work

Since implementing the new document production capabilities, the publications effort within the Computing Center has broadened considerably. In addition to user's guides and the monthly newsletter, the work now includes educational materials, proposals for developmental projects, systems documentation, information bulletins, and technical papers. The service is designed to meet the internal needs of the Computing Center and the University's Office of Systems Development.

The expanded scope of effort and the extended service capability accentuates certain problem areas. Document production now involves interaction and coordination among several groups not previously involved. This includes the author, the editorial and technical staff of Publications Services, and the trained operators who enter both the raw (original) text and subsequent updates.

One of the major problems was to develop procedures to keep track of the various copies of a document generated during its phased development. Another problem involved the routing of this information back and forth to the editorial staff, the author, and the text-entry personnel. These problems seem to have been resolved satisfactorily, judging from the performance of the system at the present time.

While serving the internal text processing needs of the Computing Center in producing documents of various types, the effort now has additional objectives. The objectives are to develop procedures and gain experience so that Computing Center personnel can assist others in utilizing interactive computerized text-processing techniques.

One of the major efforts of the Computing Center has been to develop procedures which can be utilized by secretaries and clerical personnel. The intent is to help them avoid the inadvertent losses which typically occur when inexperienced people start using the computer. In this regard, many command-procedures have been developed to simplify the use of the Text-Editor and to perform many of the repetitious functions involved in text-processing.

A special user's guide explains how to utilize the available text-processing capabilities. A computerized Programmed-Instruction (PI) course was developed whereby a secretary can, with little help, train herself to use the system. Seminars have been offered to teach people to use the Text-Editor. The computerized PI-course has even been used by the Woodrow Wilson Rehabilitation Center in Fishersville, Virginia to train disabled persons for employment in the data processing profession.

Implementation Problems

Many delays were experienced in actually getting TSO implemented and stabilized. An entire year was lost in implementing the necessary support for publications production. Nevertheless, concurrent to the stabilizing of TSO systems availability, technical personnel were exposed to the new software and developed enhancements to the system. The training of operators to do the text-entry function also consumed considerable effort over the first year of TSO availability.

The monthly newsletter, which had been a typewriter production previously, was the first task to use the Text-Editor. The extension of the availability of text-processing capabilities to other types of publications necessitated development of a raft of administrative and clerical procedures. Standards had to be developed for the text-entry personnel, and they had to be trained in these procedures. Follow-up procedures had to be devised and implemented. The mechanism for the introduction of essential graphics had to be considered, and additional equipment had to be evaluated and installed. Concurrent software development and operator training problems were encountered.

The procedures currently used to produce publications within the Computing Center were developed mostly within the previous two years. This includes recognition of the major components of the service to be offered and the sequence in which they were to be implemented. However, working out the details and imparting an understanding of the procedures among all the personnel involved was a perplexing chore at times. Consistent, coordinated follow-up and feedback in the phased publication activities was particularly difficult.

Once the production procedures were implemented, and offered on a formal basis, additional assistance had to be provided in the form of personnel capable of functioning in an editorial capacity. The objective in providing such assistance was two-fold, to minimize the effort skilled technical personnel must devote to the development of various types of publications and to produce publications of a consistently higher professional quality. Thus, it was necessary to obtain someone able to do proofreading, rewriting, and grammatical editing as well as recognize consistency with established Computing Center policy and the accuracy of the technical information presented.

Fortunately, within the Computing Center, a person was available with a degree in the languages who was also being trained to become a computer programmer. This person became the mainstay of the editorial staff and was used to develop various aids to assist authors of technical publications, and to train other members of the User Services technical staff to function in the editorial environment.

Also within the User Services Division, a group of keypunch operators were employed to service the keypunching needs of the Computing Center, members of the faculty, and two administrative offices. With the advent of interactive computing and the continued development of on-line databased administrative systems, the keypunching function (data-entry) experienced diminishing activity. Meanwhile the text-entry activity increased in demand. This situation provided an opportunity to combine the two operations with increased efficiency and productivity while minimizing the overall cost. The two functions were considered to be servicable by a single type of device under control of an operator knowledgeable in both data-entry and text-entry procedures and requirements.

Considerable work has been done in evaluating equipment and in developing software to support the combined function. While meeting with some success, the ideal hardware to facilitate this combined operation most effectively still has not been obtained. Thus, the Publications Department has gone through an evolutionary development to offer the capabilities now available.

Current Production Procedures

In the publication services offered by the Virginia Tech Computing Center several different components or activities are involved. The activities include consultation, raw-text generation, raw-text entry, copy production, editing, technical review, text updating, and archive maintenance (Figure 5). Many of these activities, of course, may be repeated numerous times when circumstances dictate.

Formal consultation between the author and Publication Services personnel may occur twice in the development of a publication. An optional preliminary consultaton seeks to help define the organization, the content, special requirements, and the type of service to be provided. Following the preliminary consultation, the author develops the raw-text which may be handwritten, typed, or dictated. Another consultation is conducted when the work is submitted in rough-draft form. At this time, a work order is initiated that spells out the editing services to be performed such as technical only, style only, a combination technical and style, or no editing at all. Due dates are discussed and agreed upon, and needed graphics support is discussed and provided for.

The raw text, regardless of the form, is entered into the system. Format commands are entered with the text to indicate margin control, line spacing (single or double), page numbering, printing with or

DOCUMENT PRODUCTION ACTIVITIES

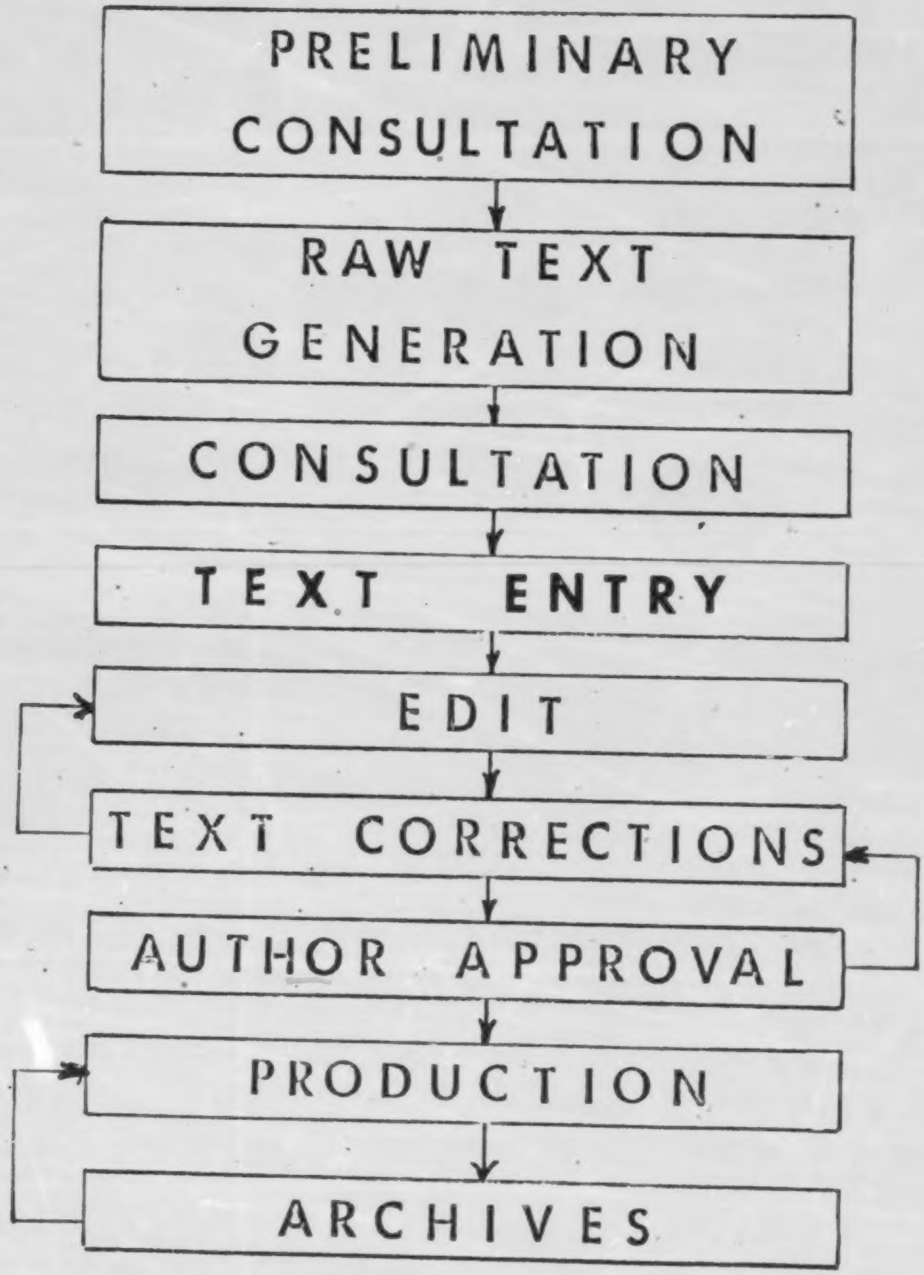

PRELIMINARY CONSULTATION
RAW TEXT GENERATION
CONSULTATION
TEXT ENTRY
EDIT
TEXT CORRECTIONS
AUTHOR APPROVAL
PRODUCTION
ARCHIVES

FIGURE 5

without line numbers, centering of lines, page headings, special spacing, text indentation, and paragraph indents. Additionally, specifications can be given for the top and bottom margins, the line length across the page (up to 120 characters), and the number of lines (length) per physical page. A number of additional commands can be used to control output formats for special purposes. Output can be printed on a high-speed line printer or a typewriter-like terminal, or displayed on a video terminal.

After the initial draft is obtained from the system, a copy edit may be performed to make the document readable and concise, by applying grammatical and formatting corrections. The style editing is performed by the Publications Department's editor. Technical editing is done by the technical staff of the User Services Division.

After this, the indicated corrections are entered into the system, updating the existing copy of the document stored on the computer. All intermediate copies of the publication are produced as line-numbered drafts to facilitate both editing and subsequent corrections. This is possible since each line of text is assigned a line number as it is entered. The copy is produced on large-size computer paper to provide plenty of space for the editing operation. A clean line-numbered draft is produced and submitted to the author for proofreading. The author verifies the information content and its technical accuracy. Based upon the author's corrections, a final draft is produced for his approval. Upon receiving the author's approval, the publication goes into production.

Once the final draft is obtained, publication of the finished document can follow three different routes: copies may be produced on a high-speed line printer; they may be reproduced photographically and collated in an administrative copy-center; or they may be produced at the campus printing plant. Once the production phase has been completed, the publication is transferred from on-line storage to a type of off-line storage medium.

Available Staff

The User Services function at Virginia Tech currently is staffed by three managers and 16 technical personnel. All of these, in one degree or another, are involved in supporting the publications (information) function. The three managers primarily devote their efforts full-time to either Publication, Consulting, or Educational Services. In addition, three members of the technical staff are committed half-time to management of the functions of the User Services Division. Thus, 14 full-time equivalent personnel are available to carry the brunt of direct services to users, including the editing and technical assistance function of Publications Services. These personnel constitute a pool of technical expertise and are drawn upon to fulfill all the functions of User Services at Virginia Tech (Figure 6).

To illustrate the sharing of the technical staff of the User Services Division, the major accomplishments of the fiscal year 1972-73 will be reviewed (Figure 7). Under administrative support, 85 separate programming projects were undertaken and completed for 10 university administrative offices. In the educational area, 44 non-credit courses were taught on 18 different topics, 2 video tapes were produced and one computerized Programmed-Instruction course was developed.

In the publications area, efforts yielded 50 different publications. Five systems guides were completely re-written, eight were extensively updated, and five completely new ones were written. Other publications include a catalog of publications available from the Virginia Tech Computing Center, the proceedings of a computer conference sponsored by the Computing Center, five technical papers, five proposals, two reports, seven system applications documents, and five sets of standards guides. Twelve issues of a much-enhanced monthly newsletter were produced.

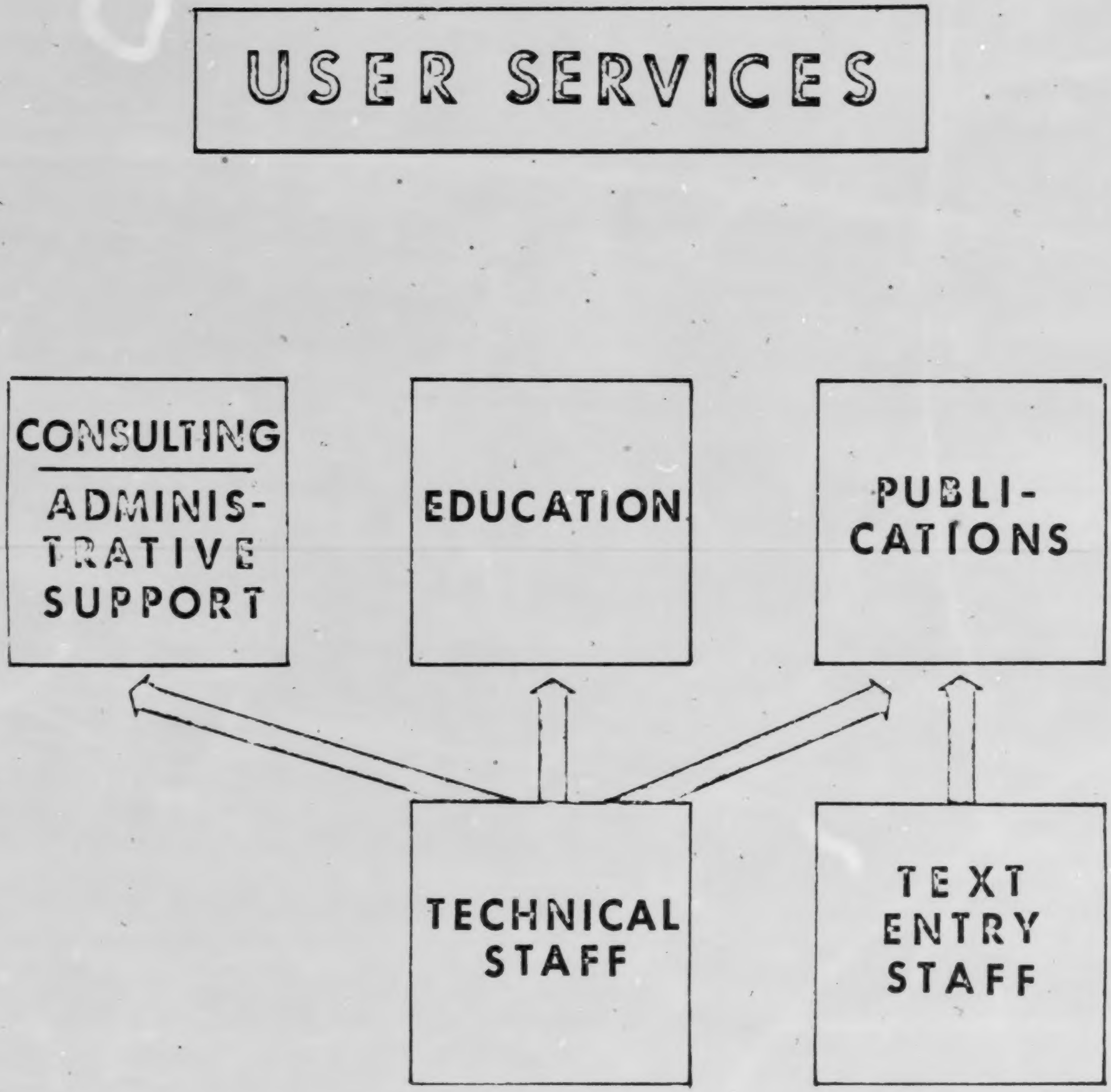

USER SERVICES
CONSULTING
ADMINISTRATIVE SUPPORT
EDUCATION
PUBLICATIONS
TECHNICAL STAFF
TEXT ENTRY STAFF

FIGURE 6

ACTIVITIES 1972-73

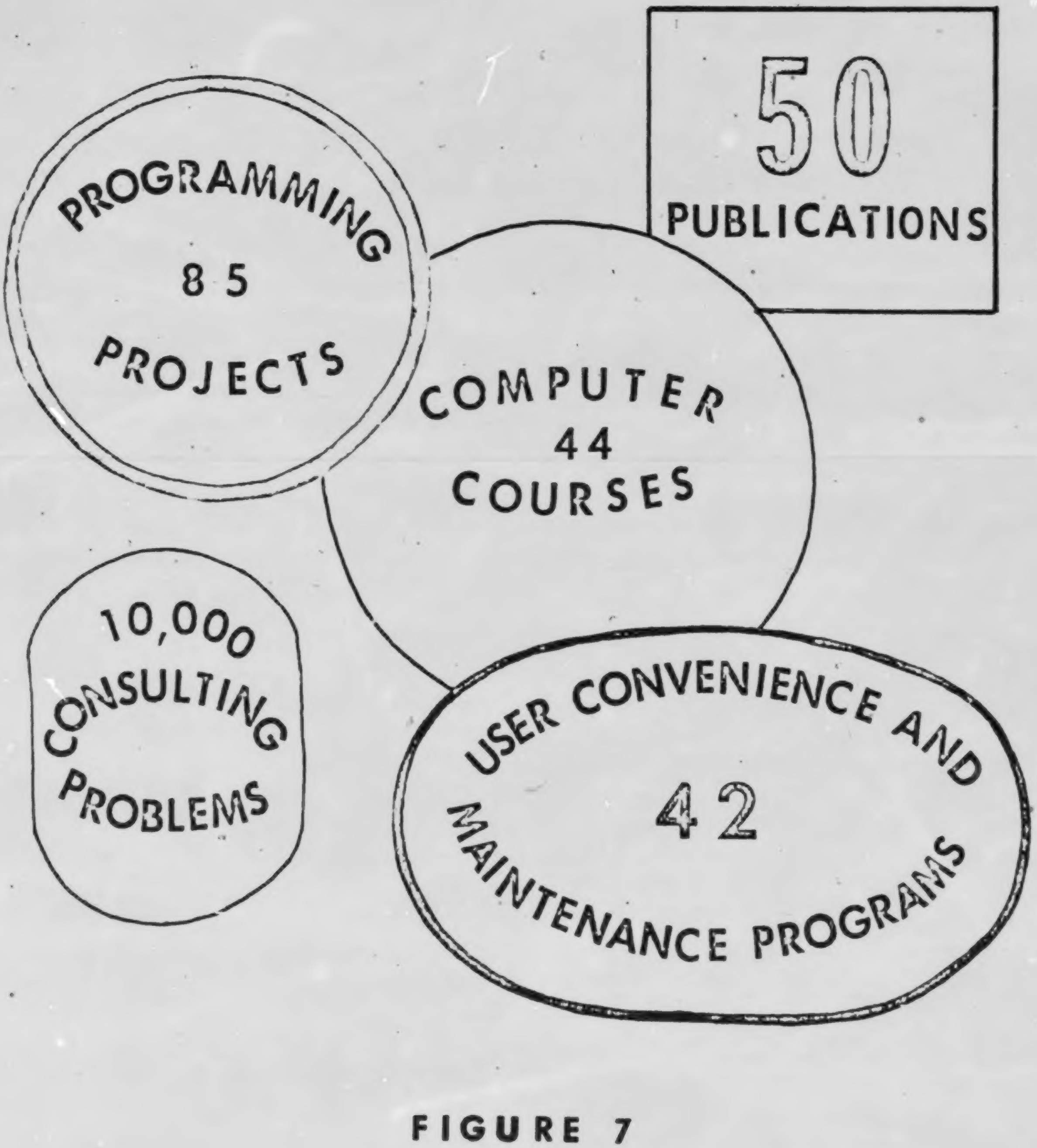

50
PUBLICATIONS
PROGRAMMING
85
PROJECTS
COMPUTER
44
COURSES
10,000
CONSULTING
PROBLEMS
USER CONVENIENCE AND
42
MAINTENANCE PROGRAMS

FIGURE 7

In the consulting area, assistance was given on over 10,000 individual problems (averaging 40 per day). Of these problems, only 60 were referred to others for solution. Seven new applications programs were added to the systems library, 10 programs already in the library were updated, 7 new utility programs were added, and 10 TSO Disk-Management Commands were developed. In addition, the conversion from IBM 2314 disk storage to IBM 3330 disk storage was effected, with minimal inconvenience to users. Four special projects in support of academic departments were undertaken while three specialized programs were developed to facilitate more effective management of the user services function.

Hopefully, the production statistics just cited convey an appropriate conception of the magnitude of the demand for a variety of services imposed on a relatively small User Services technical staff at Virginia Tech. The demands are diverse, requiring an extremely broad range of skills. The only effective means of meeting the diversity in demand for service is through a technical staff representing the entire spectrum of skills required to provide necessary services.

Potential of Text Processing at Virginia Tech

Computerized text-processing techniques have great potential at Virginia Tech, as indeed they should at any similar institution. Computerized text-processing offers the means to produce, efficiently and economically, many of the publications typically produced on a university campus. These include catalogs, bulletins, time tables, directories, schedules, reports, proposals, handbooks, technical papers, newsletters, and high-volume personal correspondence. Where highly professional print-quality is required, the present computerized techniques for text-processing may not be suitable. Nonetheless, except where esthetics are of prime concern, certain of the above mentioned publications do lend themselves to production by methods currently available with Virginia Tech's Text-Editor.

The future of the publications production effort on any university campus is likely to continue to employ a variety of techniques and equipment. For the immediate future, use of "word-processing" equipment is likely to increase at a faster rate than computerized text-processing techniques, simply by virtue of the already-established acceptance and momentum in this area. The computerized techniques are not likely to ever displace the daily routine of personal single-copy letters, memoranda, and other low-volume typewriter productions. However, the use of computerized text-processing should become one of the predominant methods utilized as scon as the power of such methods becomes more widely appreciated.

The text-processing system at Virginia Tech offers numerous performance advantages. The productivity of a typist can be increased by as much as 4-to-1 depending upon the complexity of the publication and the number of times it ordinarily would pass through the re-typing cycle. The time savings for an author can vary from 20 to 50 per cent depending upon his writing ability. The document production schedule can be shortened significantly if it is given a high enough priority. The finished document will have fewer errors since text, once in final form, is not subject to error through retyping. Obtaining additional copies of the publication is convenient regardless of the procedure utilized in final publication.

Up to the present time, the efforts within the Computing Center have been primarily devoted to developing management techniques, enhancing the software, and developing the control necessary to effectively utilize the available procedures. Only in selected instances have the techniques been promoted within the University community for use by others. The situation now permits a more active promotion of the use of these capabilities. The only remaining concern is the magnitude of the additional demand which reasonably can be serviced by the

University's computational facilities. The demand can be controlled but, if promoted and permitted to develop at an accelerated rate, the additional load could saturate the computational capabilities presently available.

The Computing Center personnel are continuing their investigation of additional enhancements to extend the capabilities of the Text-Editor and the quality of publications. This includes investigations of the typesetting equipment currently available; consideration of the possible use of computer-output microfilm equipment; evaluation of the possible use of non-impact printers; development of aids for authors to facilitate the use of dictation equipment to increase the productivity of authors in developing publications; special writing workshops; seminars on the use of text-processing techniques; and development of user's guides to facilitate publications production. Thus, the future of computerized text-processing holds great promise at Virginia Tech, particularly within the Computing Center. Certainly, it will receive increasing usage elsewhere on campus as more people become familiar with the capabilities available.

OS-OASIS CONVERSION

Roger C. Lauzon
Manager, Systems Programming
Computer Centre
University of Windsor

OS-OASIS CONVERSION

INTRODUCTION

The following paper is a technical presentation on the OS-OASIS conversion project. I will begin with a brief history of OASIS for those people not familiar with this data management system. A comparison will be made between the present DOS-OASIS structure and that designed for the OS version. The three main program segments of OS-OASIS - Master Task, Batch Monitor and On-Line Monitor - will be discussed with a very brief description of their main functions. This will be followed by an up-to-date progress report on the above three program segments, the creation of an OS data base and an OS version of OASIS. Other areas covered include documentation standards, the personnel involved, and plans concerning project completion and the distribution of the system.

HISTORY OF OASIS

In 1968, Project Info at Stanford University received a Ford Foundation Award to develop an information system. The following year the system design for OASIS (On-Line Administrative Information System) was adopted. A decision was made to run

5.9

this new information system under IBM's Disk Operating System (DOS). Even before the DOS-OASIS system became operational, institutions began to request an OS version of OASIS. In June 1972, the OS-OASIS Conversion Design Specification manual was produced by Project Info to assist another institution in programming the conversion to OS. Project Info contracted an IBM Service Engineer to prepare this manual. The Computer Centre at the University of Windsor reviewed this manual and a decision to undertake the conversion was made approximately one year ago. Since that time, the DOS system has extended its support to 3330 disks besides 2314's, and work is under way to convert terminal support from Sanders 720's to IBM 3270's.

DOS-OASIS VS OS-OASIS STRUCTURE

DOS-OASIS

The structure of the DOS system is made up of a permanently resident 2K module (POASIS) in the nucleus and two partitions, one for On-Line OASIS and the other for Batch. (Diagram 1.) One of the POASIS functions is to intercept Supervisor Calls (SVC's) and program interrupts to see if they belong to an OASIS task. If they do belong to OASIS, an appropriate action is taken, otherwise control is passed to DOS. The On-Line and Batch systems issue OASIS SVC's to communicate information to tables common to both partitions (e.g. task activity tables, log tape status tables, etc.) and to execute some common routines (e.g. register saving and restoring, enqueuing and dequeuing of resources, etc.). POASIS is also used for OASIS initialization. In this capacity it provides communication with the operator to learn the present status of the log tape to be mounted, which will record all transactions to the data base.

The On-Line and Batch partitions contain file service routines and data base and log tape input/output modules which are duplicated in both partitions. This duplication is necessary since the On-Line and Batch systems could run concurrent or separate. As a result, both partitions must have facilities to communicate with the data base and log tape and have similar file services. On-Line OASIS also contains routines for terminal input/output, a scheduler to control all on-line tasks (terminals) and several 2K module areas into which are loaded, upon request, terminal application modules. In the Batch partition, the Batch-OASIS system modules are linked to the user's application code, which instructs the system to update the data base, query certain files, or execute whatever service it wishes and has authority to perform.

OS-OASIS

The OASIS system under OS is basically divided into three partitions - the Master Task, the On-Line Monitor and the Batch

Monitor. (Diagram 2.) These three partitions will occupy a substantially smaller amount of core than the DOS system; also, control over the data base and log tape is improved. The Master Task will now contain the file services and the data base and log tape input/output routines which were duplicated in the DOS system. The OASIS initialization process and some common routines located in the DOS nucleus in POASIS also make up part of the Master Task. This Master Task will be a system task under OS and will have the highest dispatching priority among the three OASIS partitions.

The next highest dispatching priority partition will be the On-Line Monitor. The on-line partition will no longer do data base and log tape input/output, but will continue to perform polling and input/output functions for the terminals. The scheduler controlling on-line tasks and the module areas containing application modules will be left in the on-line partition with the same basic functions. To execute the file service routines and perform input/output to the data base and the log tape, an interface to the Master Task is necessary.

The Batch partition will consist mainly of the user application program. The Batch Monitor, which includes an interface to the Master Task and some routines common only to certain batch applications, will be linked to the user program. The user program must make an initial call to the entry point 'OIPL' in the Batch Monitor to initiate this job as an OASIS task and set up the proper tables. Subsequent calls for OASIS services will be handled by the included routines or the interface to the Master Task.

Besides the above coding in the three partitions, OASIS requires two type I SVC's in the OS nucleus and a table in MVT's Link Pack Area or in MFT's Resident Re-entrant Module Area. One of the SVC's (Linkage SVC) is used to notify the Master Task of a request for service from the on-line monitor's or batch monitor's interface. The other SVC (Protect Key SVC) is called by the On-Line Monitor to manipulate storage and PSW protect keys to implement the OASIS storage protection feature. The table in the LPA(MVT) or in the Resident Re-entrant Module Area (MFT) will contain OASIS system status flags referred to by both monitors, and address slots for event control blocks (ECB's) used to synchronize service requests by the monitors and input/output functions to the data base and the log tape by the Master Task.

OS-OASIS PROGRAM SEGMENTS

Master Task

The Master Task will operate as an OS System Task in a high dispatching priority partition or region. A System Task is a privileged program that operates as if it is a part of the operating system (executes with a protect key of 0). This special program should run just below the OS Reader/Interpreter or Writer in an OS system or below HASP in a HASP/OS system. In this way the Master Task, which is very I/O-bound, will not be interrupted by more C.P.U.-bound programs.

A combination of new code and conversion of existing DOS-OASIS routines make up the OS-OASIS Master Task. The main functions are initialization, control, termination, service routines, log tape and data base I/O and the enqueue/dequeue routines. (Diagram 3.) The initialization routine initializes the OASIS system. This module opens the data sets for the data base, inquires from the console operator the status and options for the log tape and/or user tape, sets flags appropriately in the table in the LPA or in the Resident Re-entrant Module Area to indicate Master Task initiation, and then, after communicating with the operator, it overlays its coding with newly-built tables. An exit is made from here to the control routine. This routine contains a multiple WAIT issued against a list of ECB addresses (in the LPA table), and code necessary to analyze which ECB was POSTed and where control should pass. For example, when one of the monitors would like to execute a Master Task service, an ECB is POSTed in the LPA table which wakes up the Master Task (makes it dispatchable) and then transfers control to the appropriate service. Upon completion of the service, control is passed to a service completion routine which returns to the calling monitor.

The service routines are converted DOS-OASIS "file services" and "application support services" now resident in the Master Task. These services (e.g. PASSWORD, ADDSEG, etc.) are called by user application programs from the monitors or from other service routines to manipulate data in the data base. To access the disk files, a data base I/O routine is used. This routine adds, deletes, and updates information in the data base files through its own channel programs. No standard OS access methods are used in data base I/O, but the OASIS channel programs support both 2314 and 3330 disk drives. When information in the data base is updated, a backup routine logs the updated transaction on a log tape via a log tape I/O routine. This is done for every update and at the beginning and end of every OASIS task, so that no information can be lost in case the system crashes unexpectedly. The log tape I/O routine also issues its own channel programs, but allows OS to do Standard Label processing.

The enqueue/dequeue function, which maintains information about the status of resources available to OASIS tasks, along with register saving and releasing functions, are routines converted to reside in the Master Task after being resident in POASIS (in the nucleus) under DOS-OASIS.

Termination in the Master Task can occur for two reasons: normal termination due to an operator request or abnormal termination as a result of some type of program check, etc. In either case, files will be closed and flags will be set in the LPA table indicating Master Task termination to the monitors. Without the Master Task running, all OASIS functions will terminate since the Master Task is a definite requirement to execute OASIS on-line or batch tasks.

Batch Monitor

The Batch OASIS partition or region is made up of the user application program written in Assembler, Cobol, or PL/1 and the Batch Monitor which controls this OASIS batch program. (Diagram 4.) A special use of the Batch Monitor occurs for data base maintenance, reconstruction and backup which will be referred to as "maintenance" programs.

In the OS-OASIS structure, more than one partition can run OASIS batch programs. The only restriction is that the program be link-edited to the Batch Monitor. The monitor takes up very little core and the batch partition will function like any other OS batch partition; (its protect key is unchanged).

The Batch Monitor is made up of functions for initialization, services interface, termination, and creation of table and data areas, along with a few routines necessary for "maintenance" programs. The first call a user application program must perform is to the Batch Monitor initialization process, the entry point 'OIPL'. This routine converts the calling program to an OASIS task by calling the backup routine in the Master Task, which will write a 'start' record for that task on the log tape. This module also modifies the return register in the OS save area to point to the Batch Monitor's termination routine, so that at the end of the application program, control will return to the monitor's termination, not to OS as a program normally would. The table in the LPA is checked to ensure that the Master Task is operational and the table and data areas for that task are created. Return is now made to the user program, which can call for OASIS services.

When an OASIS service is called, control is transferred to a correspondingly named entry point in the interface. The interface contains an entry point for every service in the Master Task which a batch program could call. The interface sets up a parameter list for that service call and transfers control to the Master Task via the Linkage SVC.

The termination routine will gain control at the end of the user program (normal end) or when a program check occurs (abnormal end). In either case, an 'end' record will be put out to the log tape via the Master Task and appropriate flags will be set in the LPA table before control is returned to OS.

On-Line Monitor

The On-Line Monitor operates as a conventional OS job with some special facilities. The protect key of the partition or region will be set to zero by the monitor's initialization routine. The On-Line Monitor modifies its protect key to implement the OASIS storage protection feature. Here is how the feature works: when a 2K program module in the on-line partition is made active by an on-line task (terminal), then the storage protect key for this 2K module is changed to the original protect key for the partition (e.g. 3 for P02). Any time an on-line task leaves the active state, then the storage protect keys for modules associated with it should be changed to zero. In this way, program modules associated with other on-line tasks are protected from inadvertent modification by an active program module. If an active program module tried to write in modules not associated with the active task, the system will generate a protection exception.

The general functions of the On-Line Monitor include initialization, control, task scheduler, services interface, terminal and terminal screen format I/O, module loader and module area, QUERY, Command Language Processor (CLP) and Terminal Report Generator (TRG) functions, and termination. (Diagram 5). I will not explain these functions in depth since the OS On-line Monitor has not been coded to date and so the functions of the current DOS-OASIS procedures may be modified. Some of the *initialization* process has already been discussed (setting protect keys); there remains the opening of data sets for the terminals, setting an initial timer interval for terminal polling and turning on flags in the LPA table. The *control* routine will contain a multiple WAIT on a list of ECB's similar to the Master Task and the *services interface* will resemble that in the Batch Monitor. The *task scheduler* scans for on-line tasks eligible to be made active and passes control to them, otherwise returning to the multiple WAIT. The *I/O routines* handle all requests for terminal I/O to IBM 3270's with their own channel programs. Modifications are now being made to the DOS-OASIS system to support IBM 3270's instead of Sanders 720's. *QUERY* and *TRG* are terminal services, whereas *CLP* handles terminal commands. The *termination* routines will be similar to the Master Task's, but will notify the backup routine in the Master Task in the event that recovery procedures must be initiated.

PROGRESS REPORT

Testing has begun on the first stage of the conversion project. This first stage includes the Master Task, the Batch Monitor and the OS Data Base Creation. With the completion of this testing, a Batch OS-OASIS system will be operational.

To create the OS data base, batch "maintenance" routines (mentioned in the section on the Batch Monitor) have to be converted and the data base input/output module must be fully tested. The testing procedures have presented us with a dilemma; we require a data base to test the system, but to create the data base, the system is necessary.

The On-Line Monitor, which controls the on-line tasks, will be designed and coded as the second stage of our conversion project. This will be done at the beginning of the new year after the Batch OS-OASIS system is fairly stable.

There are plans underway to have an OS-VS version of OASIS. Both Stanford and ourselves are looking into ways to keep the VS conversion to a minimum and, in fact, to include into the OS-OASIS system features which will be compatible to both OS and OS-VS.

A progress report on the conversion project is published occasionally to inform interested individuals of any new developments. Contact me if you would like to be placed on our mailing list.

Miscellaneous Areas

Originally, we were informed that the conversion effort would be a twelve man-month project. At the present, we now have four system programmers from the University of Windsor and one from the government of Quebec actively involved in the project. Besides the personnel, technical assistance from the Project Info group at Stanford is constantly available. To date, approximately 35 man-months have been devoted to the project.

Documentation for every newly-coded or modified module in the OS-OASIS system is strictly emphasized. This includes a description of the function of each entry point in the module, the external references called, macros used, etc. At the end of the project, documentation would include a description of the generation and internal operation of OS-OASIS, along with the writeups on the system modules.

A project of this size and complexity requires close management and strict control by the project leader. It is a very difficult task for programmers to constantly take apart another person's coding to understand the operation of a routine, without getting depressed or going off on tangents. Constant reminders must also be given to keep documentation standards up-to-date.

A question most often asked is why did we decide to convert OASIS to OS? After an extensive evaluation of other data management systems, including a six month preparation and demonstration of IMS, the enormous expense of $10,000 to $20,000 a year or more on a badly needed data management system was more than we could afford or justify. The OASIS system had all the facilities of the other data management systems; it was free, but had to be converted. The need of a data management system and the economic situation the way it is in our university, not unlike other educational institutions, were the deciding factors to undertake the conversion.

The Batch OS-OASIS system should be relatively stable by early spring, since testing is now being done. We should begin testing the On-Line system by the summer of 1974, and it should be completed by the end of the summer if not sooner. At that time, an unsupported OS-OASIS version can be distributed free of charge to educational institutions. Ideas to supply full support and a minimum support plan (for those installations with ample support staff) have been discussed, but an estimate of the cost to supply different levels of support has not been finalized.

DOS - OASIS STRUCTURE

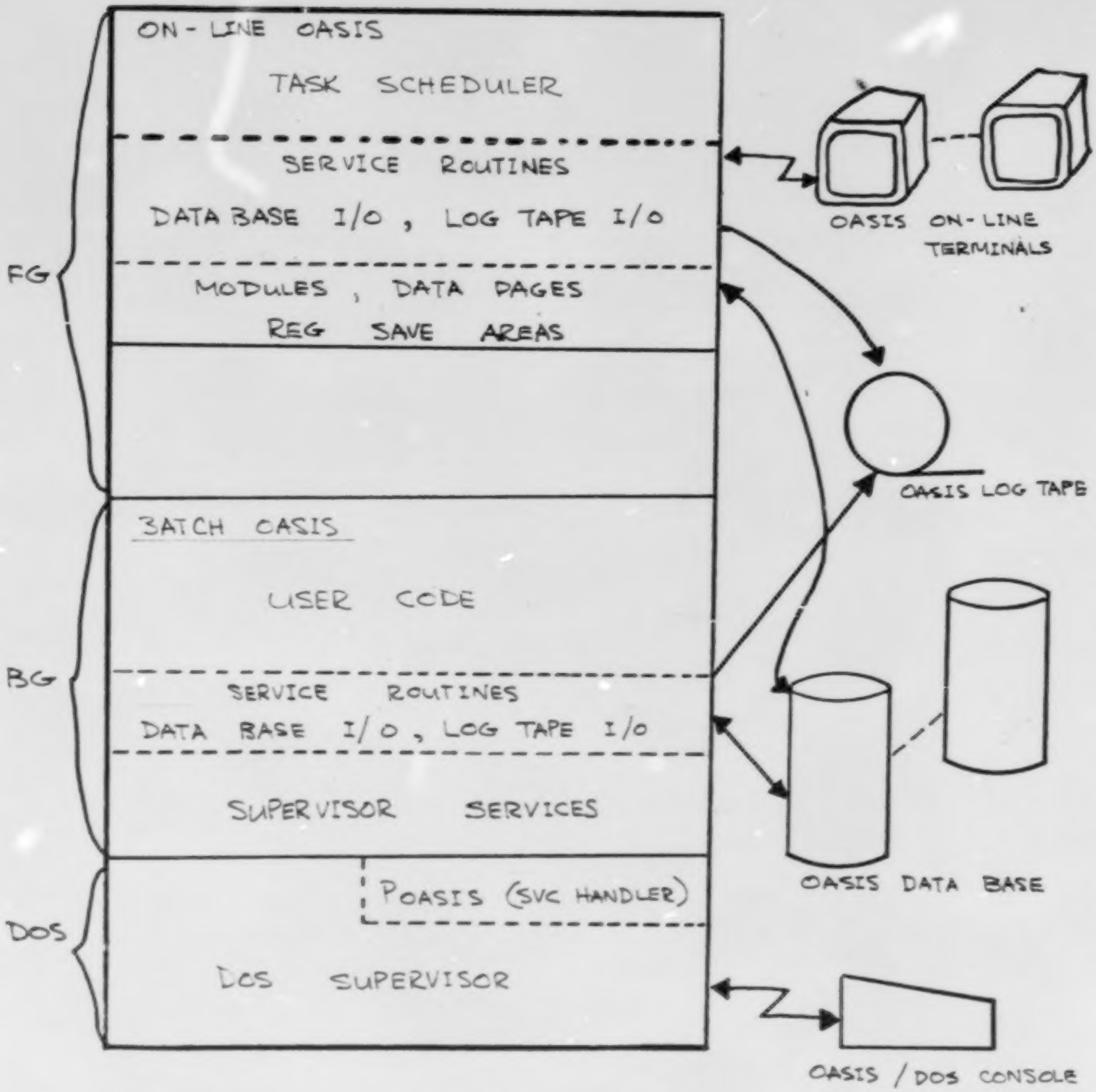

ON-LINE OASIS
TASK SCHEDULER
SERVICE ROUTINES
DATA BASE I/O , LOG TAPE I/O
MODULES , DATA PAGES
REG SAVE AREAS
FG
BATCH OASIS
USER CODE
SERVICE ROUTINES
DATA BASE I/O , LOG TAPE I/O
SUPERVISOR SERVICES
BG
POASIS (SVC HANDLER)
DOS SUPERVISOR
DOS
OASIS ON-LINE TERMINALS
OASIS LOG TAPE
OASIS DATA BASE
OASIS / DOS CONSOLE

DIAGRAM 1

OS - OASIS STRUCTURE

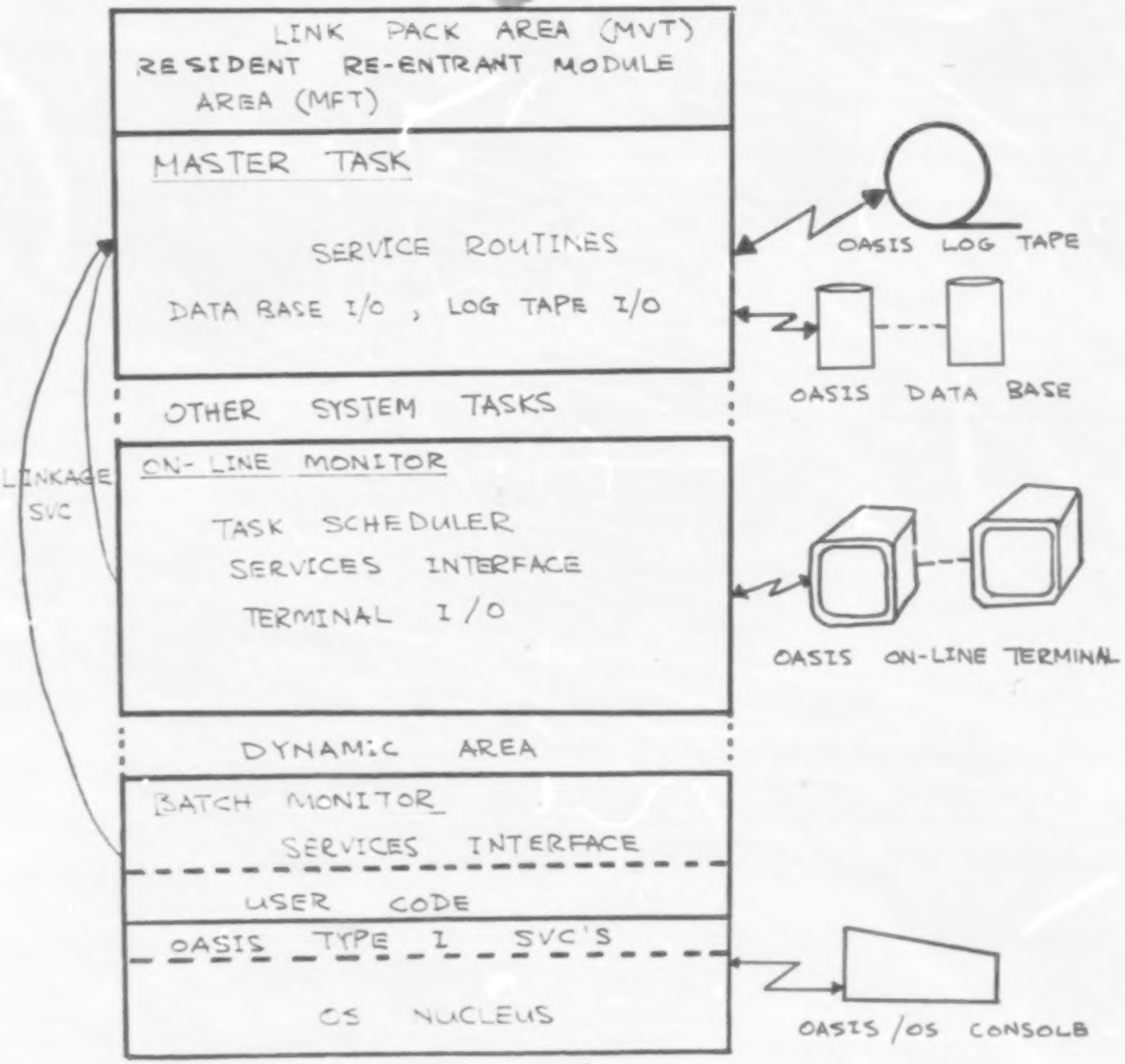

LINK PACK AREA (MVT)
RESIDENT RE-ENTRANT MODULE AREA (MFT)
MASTER TASK
SERVICE ROUTINES
DATA BASE I/O , LOG TAPE I/O
OTHER SYSTEM TASKS
ON-LINE MONITOR
TASK SCHEDULER
SERVICES INTERFACE
TERMINAL I/O
DYNAMIC AREA
BATCH MONITOR
SERVICES INTERFACE
USER CODE
OASIS TYPE I SVC'S
OS NUCLEUS
LINKAGE SVC
OASIS LOG TAPE
OASIS DATA BASE
OASIS ON-LINE TERMINAL
OASIS /OS CONSOLE

DIAGRAM 2

MASTER TASK

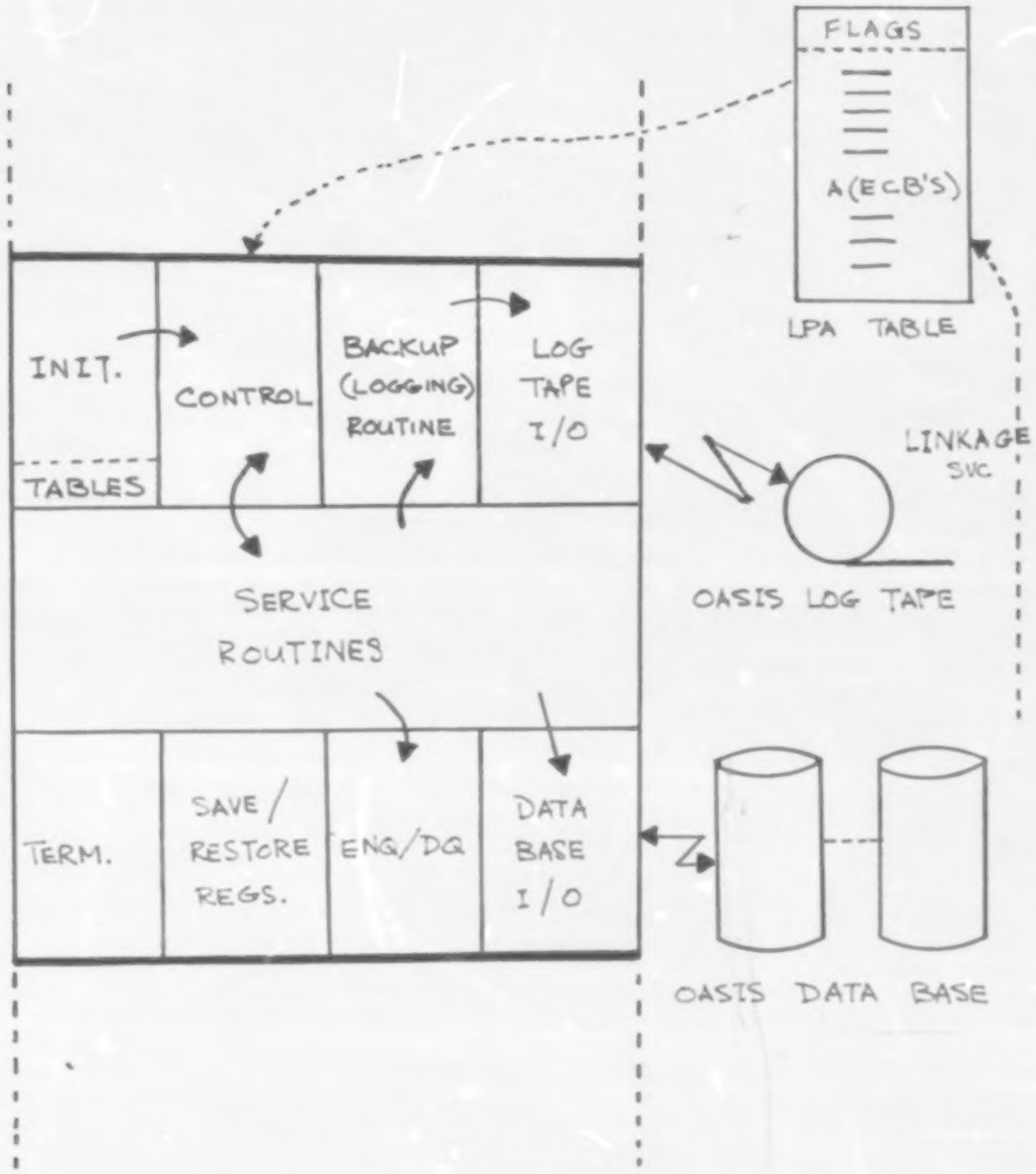

FLAGS
A(ECB'S)
LPA TABLE
INIT.
TABLES
CONTROL
BACKUP (LOGGING) ROUTINE
LOG TAPE I/O
LINKAGE SVC
SERVICE ROUTINES
OASIS LOG TAPE
TERM.
SAVE / RESTORE REGS.
ENQ/DQ
DATA BASE I/O
OASIS DATA BASE

DIAGRAM 3

BATCH MONITOR

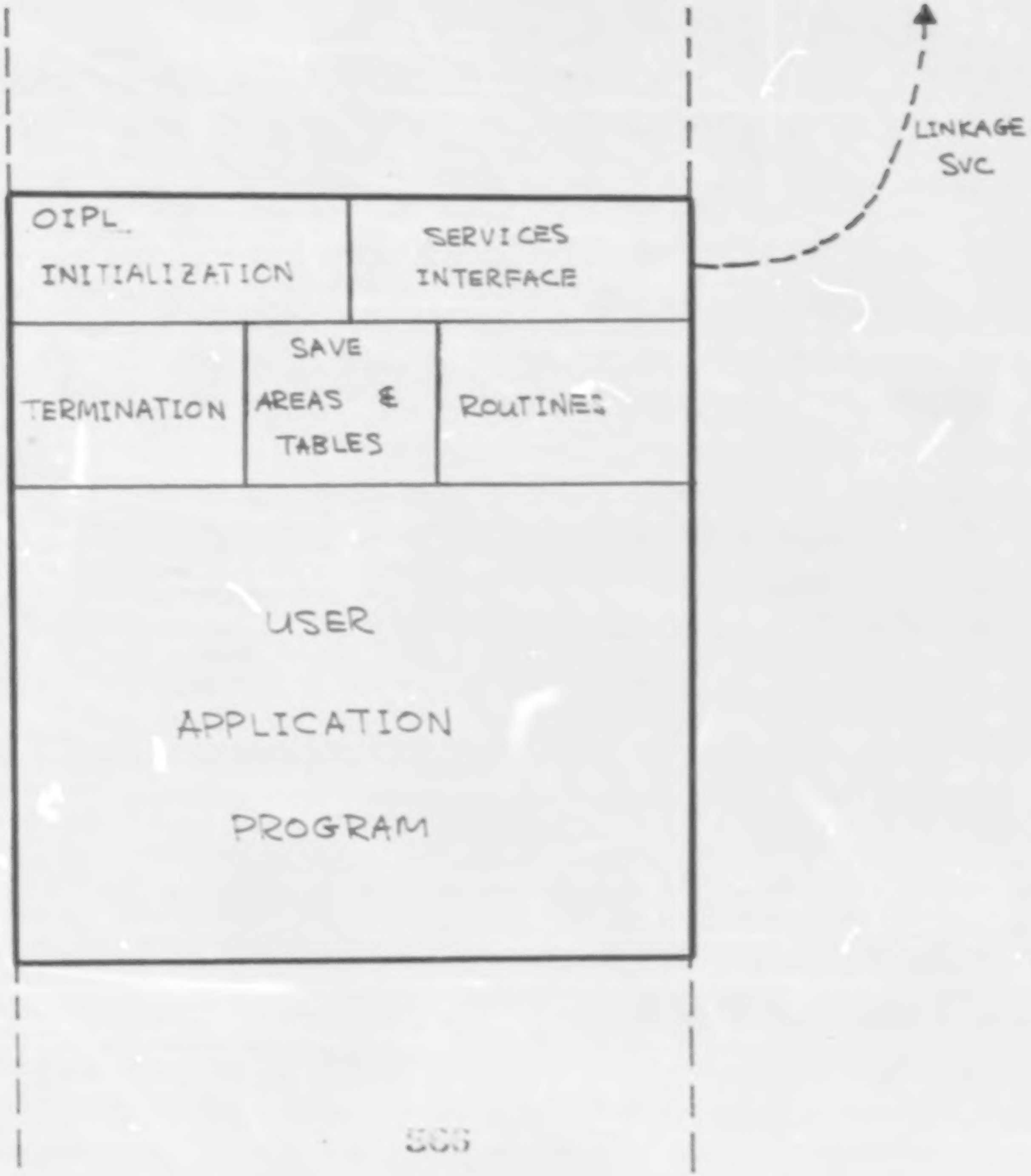

OIPL
INITIALIZATION
SERVICES INTERFACE
LINKAGE SVC
TERMINATION
SAVE AREAS & TABLES
ROUTINES
USER APPLICATION PROGRAM
SCS

DIAGRAM 4

ON-LINE MONITOR

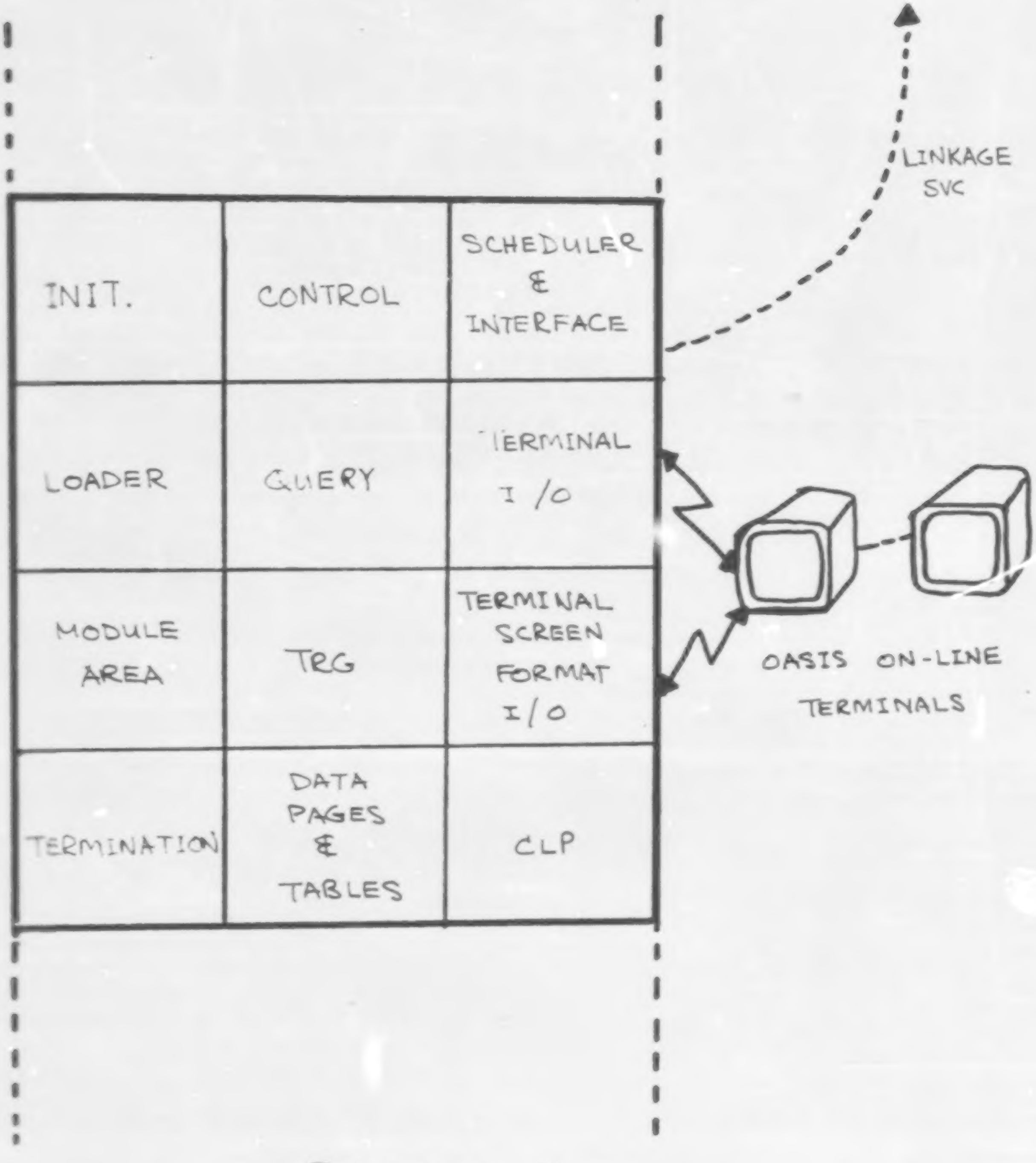

INIT.
CONTROL
SCHEDULER & INTERFACE
LINKAGE SVC
LOADER
QUERY
TERMINAL I/O
MODULE AREA
TRG
TERMINAL SCREEN FORMAT I/O
OASIS ON-LINE TERMINALS
TERMINATION
DATA PAGES & TABLES
CLP

DIAGRAM 5

SOFTWARE DESIGN AND ARCHITECTURE

James W. Boyd
Director of Data Processing
College of DuPage

SOFTWARE DESIGN AND ARCHITECTURE
AT
COLLEGE OF DUPAGE GLEN ELLYN, ILLINOIS

James W. Boyd
Director of Data Processing

The College and Its Community

The College of DuPage is a two-year community college located in the suburban community of Glen Ellyn, Illinois, thirty miles west of Chicago. The College has grown from an initial enrollment of 2615 students in 1967 to 9996 for the current quarter. It offers a wide variety of educational programs in the three areas of transfer, occupational and general studies. Over 106 different educational plans are available, with individualized study an alternative in many subjects.

The area serviced by the College covers almost all of DuPage County and small sections of Cook and Will counties. There are thirty five communities with a total population of over 600,000. It is a rapidly growing area with the population expected to double over the next eight to ten years. Student population for the College is projected to grow to 20,000 by 1980. There are 53 different high school and elementary school districts in the area served by the College.

Data Processing Department

The Data Processing Department is a service organization designed to support the policies of the Board and the President of College of DuPage in the areas of administrative and instructional uses of computer equipment. It is potentially a source of service to all functions of the College. In addition, the Department offers computer services to schools, municipalities and non-profit organizations external to the College to the extent permitted by equipment and personnel.

Data processing equipment and personnel are centrally administered by the director of Data Processing Department who reports to the President of the College.

Department Functions

In order to adequately support the general policy, the Data Processing Department is functionally divided into three basic areas:

1. Development of new systems
2. Operation of existing systems or production
3. Support of Data Processing instructional program.

1. Development of new systems involves eight distinct steps

   a. Identification of user needs--This requires contact and communication with potential users and being alert to new developments in educational data processing.

   b. Analysis and Cost effectiveness--This requires a careful analysis of present user systems and projection of cost effectiveness of developing a computerized system. It also involves a general estimate of equipment and personnel requirements and the time required for implementation.

   c. Project approval and budget allocation--This is a function of the director of Data Processing working with the Management Committee.

   d. Assignment of Priority--This is a function of the appropriate planning committee in conjunction with the Data Processing Management Committee

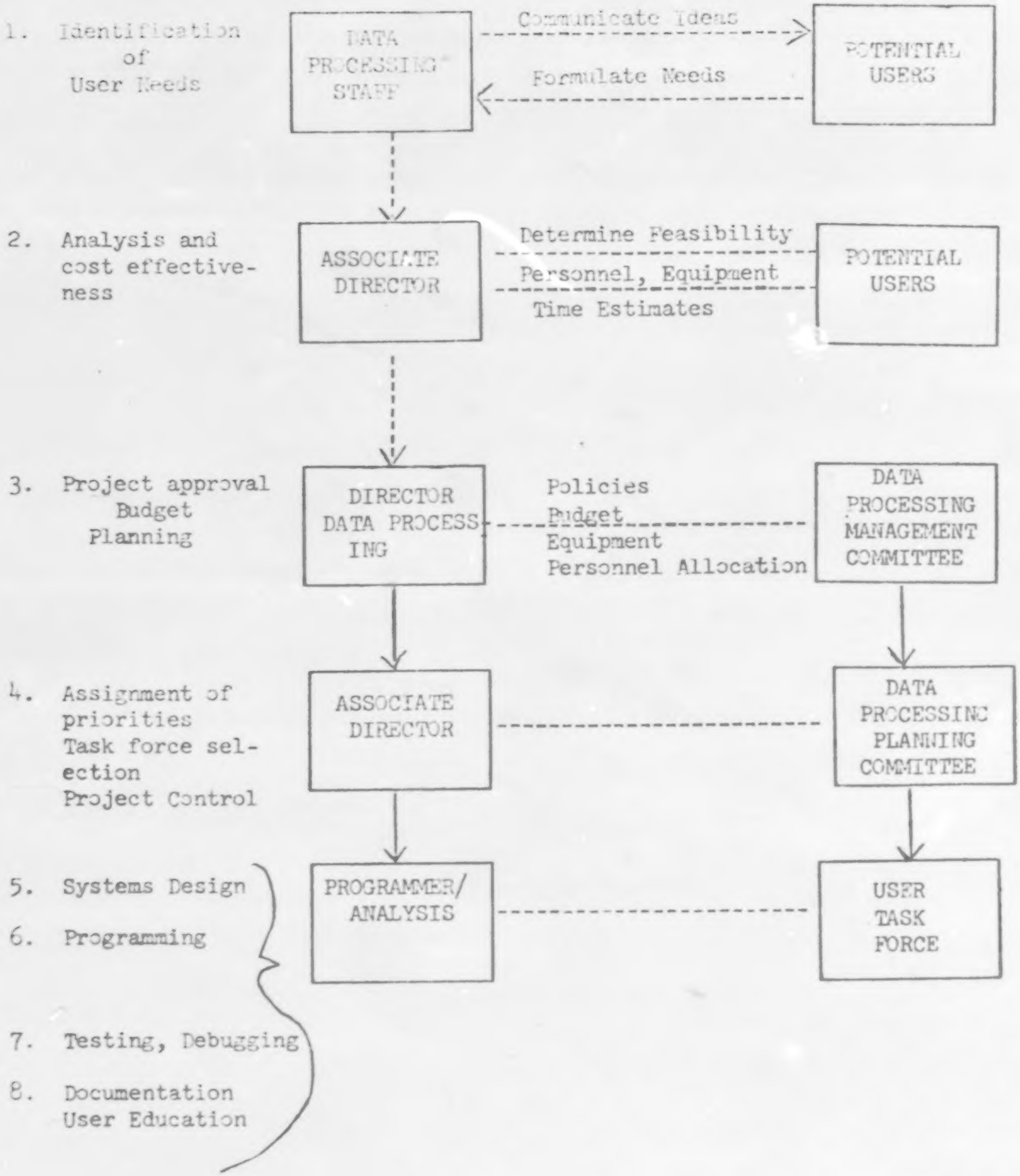
   e. Systems Design--This is a detailed development of the overall systems requirements specifying the general flow of data, the layout of files and the general logic of the system. It is a function of a user-task force working with Data Processing personnel.

   f. Programming--This involves the detailed coding, and file creation involved in the system.

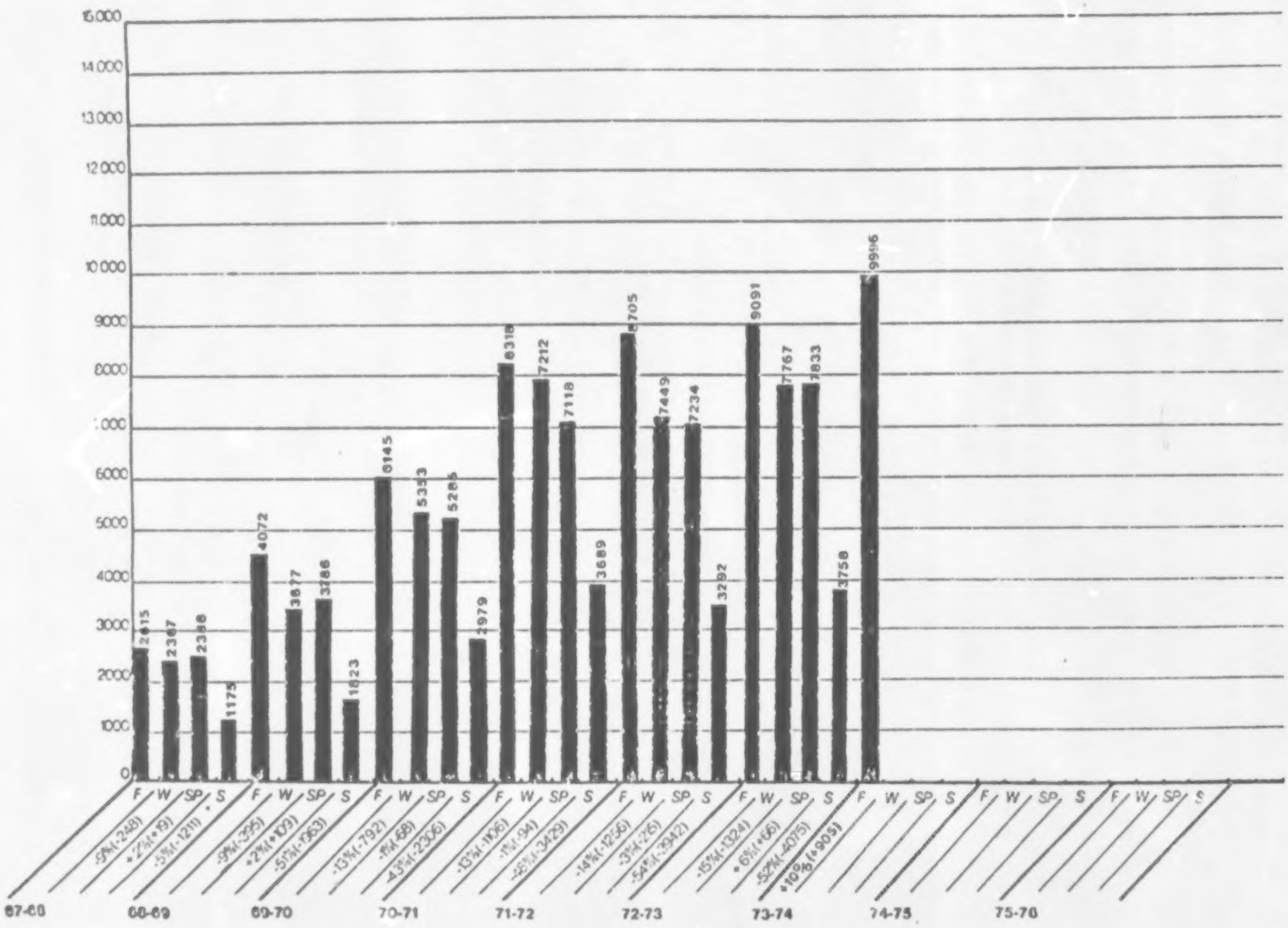
   g. Documentation and user education--This involves the development of written description and procedures of the system. Documentation must provide adequate instructions for placing the system into production, and maintaining the system. In addition, the potential user must be given adequate instructions to enable him to satisfactorily use the system.

A diagram of the development process is presented on the next page:

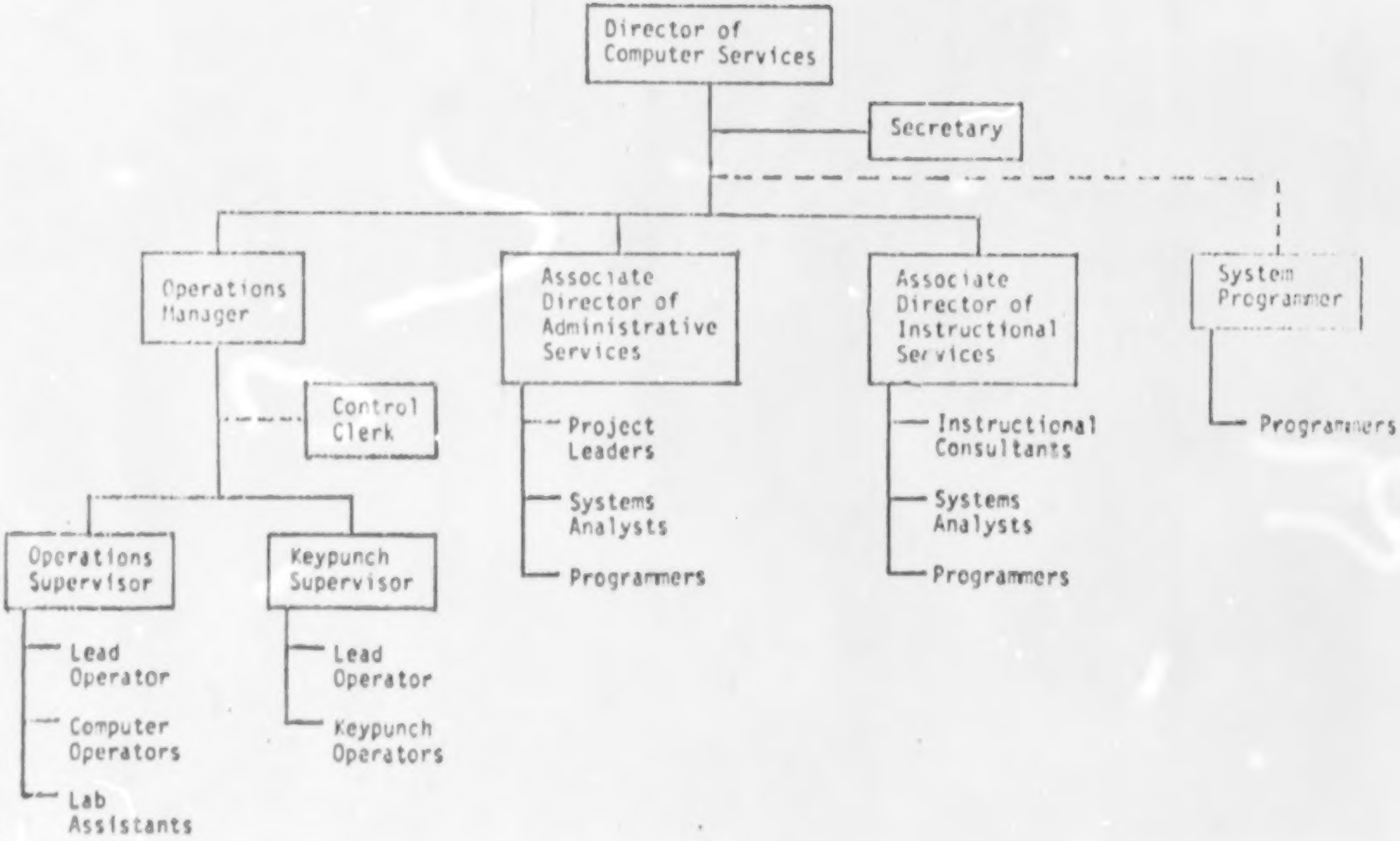
DIAGRAM OF DATA PROCESSING DEVELOPMENT PROCEDURE


1. Identification of User Needs
DATA PROCESSING STAFF
Communicate Ideas
Formulate Needs
POTENTIAL USERS
2. Analysis and cost effectiveness
ASSOCIATE DIRECTOR
Determine Feasibility
Personnel, Equipment
Time Estimates
POTENTIAL USERS
3. Project approval Budget Planning
DIRECTOR DATA PROCESSING
Policies
Budget
Equipment
Personnel Allocation
DATA PROCESSING MANAGEMENT COMMITTEE
4. Assignment of priorities Task force selection Project Control
ASSOCIATE DIRECTOR
DATA PROCESSING PLANNING COMMITTEE
5. Systems Design
6. Programming
7. Testing, Debugging
8. Documentation User Education
PROGRAMMER/ ANALYSIS
USER TASK FORCE

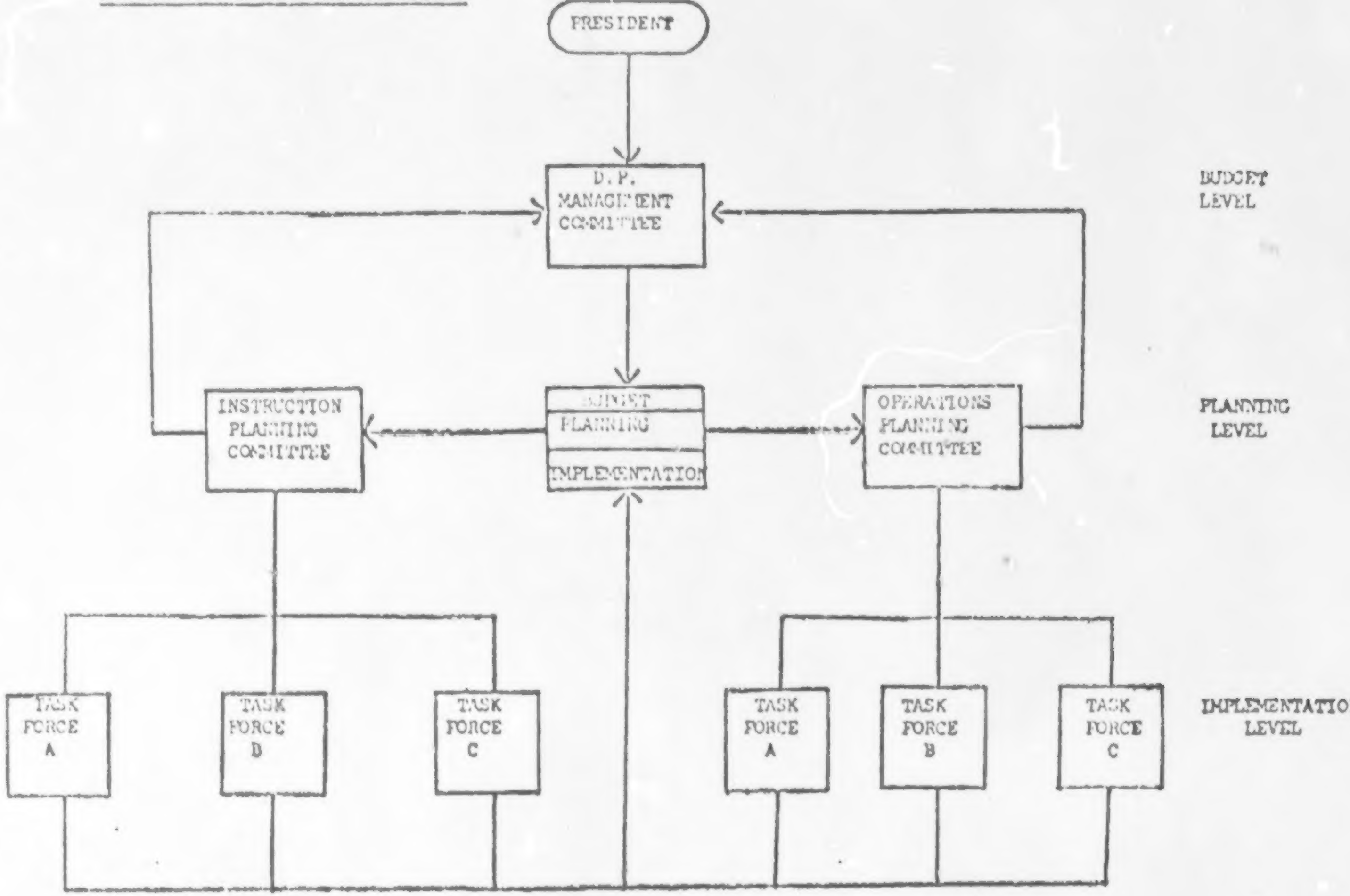
HEAD COUNT (Peak Enrollment Figures)



COLLEGE OF DuPAGE COMPUTER SERVICES


Director of Computer Services
Secretary
Operations Manager
Control Clerk
Operations Supervisor
Lead Operator
Computer Operators
Lab Assistants
Keypunch Supervisor
Lead Operator
Keypunch Operators
Associate Director of Administrative Services
Project Leaders
Systems Analysts
Programmers
Associate Director of Instructional Services
Instructional Consultants
Systems Analysts
Programmers
System Programmer
Programmers

INSTITUTIONAL PLANNING STRUCTURE


PRESIDENT
D.P. MANAGEMENT COMMITTEE
BUDGET LEVEL
INSTRUCTION PLANNING COMMITTEE
BUDGET
PLANNING
IMPLEMENTATION
OPERATIONS PLANNING COMMITTEE
PLANNING LEVEL
TASK FORCE A
TASK FORCE B
TASK FORCE C
TASK FORCE A
TASK FORCE B
TASK FORCE C
IMPLEMENTATION LEVEL

How the Computer is Used
at
College of DuPage

The College of DuPage Data Processing Department uses an IBM 360 model 40 computer to perform a variety of jobs for both internal and external users. The Data Center, located in K104, operates 24 hours each day from Sunday midnight to Friday midnight. The facilities are used for student labs on Saturday morning and are rented to the County Health Department on Saturday afternoon and Sunday morning. A tremendous amount of output is produced in printed form and through cathode ray tube visual displays.

System Configuration

The computer system at the College of DuPage is made up of the following hardware components:

1 - IBM 360/40 Central Processing Unit
1 - 1100 per minute line printer
1 - 1000 per minute card reader/punch
2 - 800 bits per inch tape drives
6 - Direct Access Storage Disk Drives
  (To become 9 drives July 1, 1973)
14 - Cathode Ray Tube (CRT) Visual/Display terminals at C/D
8 - CRT Visual/Display terminals at Willowbrook High School
6 - CRT Visual/Display terminals at York High School
6 - CRT Visual/Display terminals at Addison High School
1 - CRT Visual/Display terminal at Glencrest Junior High School
1 - CRT Visual/Display terminal at George Williams College

Data Processing Staff

The Data Processing Department has a staff of 24 people. The staff positions are:

1 - Director
3 - Associate Directors
  Operations
  Instruction
  External Services
1 - Systems Programmer
9 - Programmer/Analyst
  3 - Operations
  4 - Instruction
  2 - External Services
1 - Operations Supervisor
1 - Input/Output Controller
3 - Computer Operators
4 - Keypunch Operators
1 - Secretary

Types of Applications

Data Processing applications are divided into two general areas according to the type of interface with the computer.

1. On-line applications - These are applications which involve user interaction with the computer. Through the use of Cathode Ray Tube display terminals, information may be entered, updated or displayed. There are over 50 unique on-line functions in operation within the College of DuPage system. Among the on-line functions are the following:

    a. On-line Registration
    b. Add/Drop
    c. Student Add/Update
    d. Grade Updating
    e. Schedule Creation
    f. Schedule Management
    g. Financial Aid Display/Entry
    h. Advisor Display
    i. Budget Control
    j. Student Exploration
    k. Registration Holds
    l. Student Exploration of His Record
    m. Vocational Information Retrieval
    n. College Information Retrieval
    o. Transfer Planning
    p. College of DuPage Curriculum
    q. Local Job Information
    r. On-line Record Updating
    s. Computer Assisted Instruction

2. Batch applications - These are applications which are not interactive and usually involve printer output. All hard-copy reports are examples of batch applications. Among the many batch applications in operation are the following:

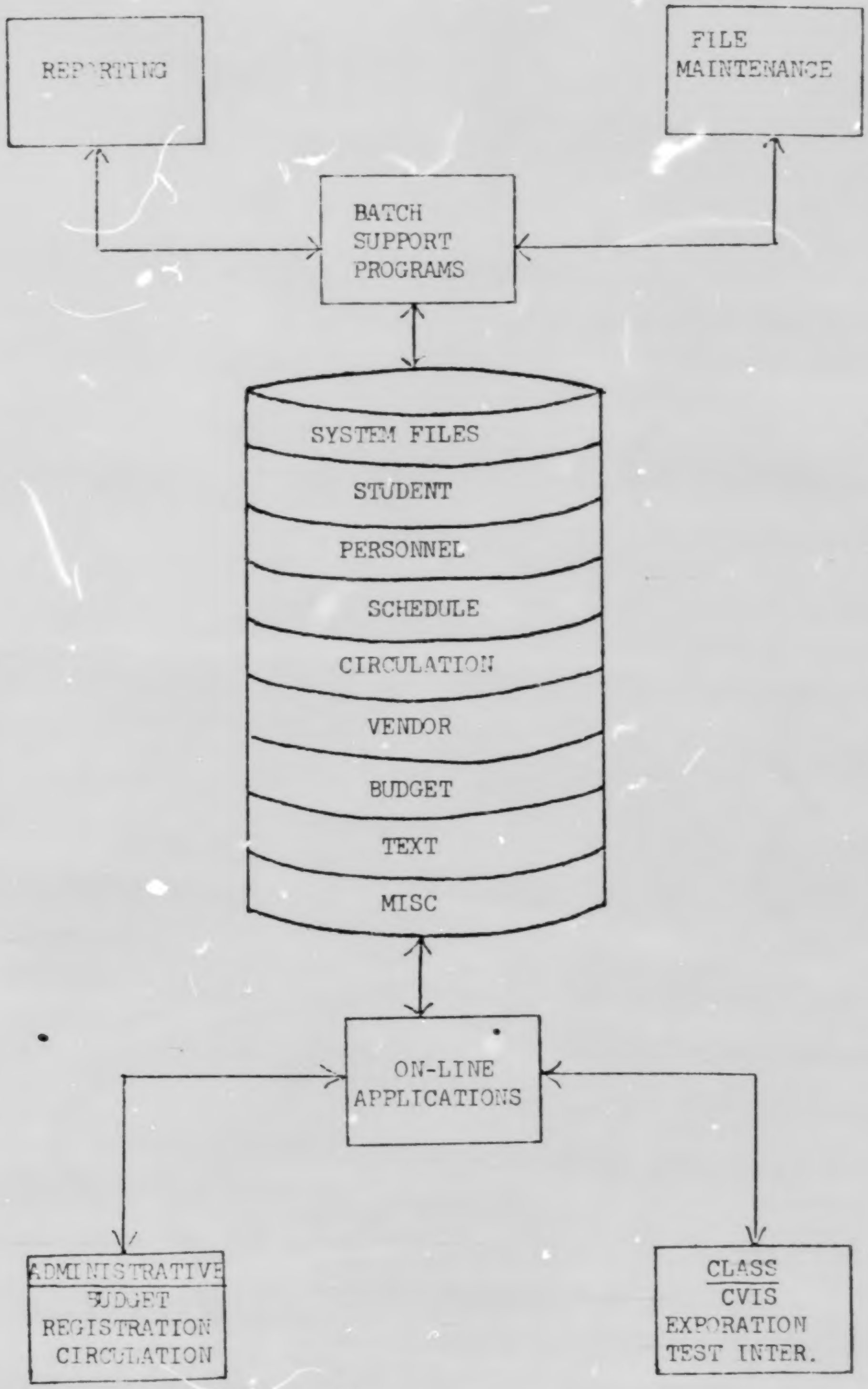
    a. Payroll
    b. Budget
    c. Grade Reporting
    d. Registration Appointment
    e. Advisor Reports
    f. Class Lists
    g. High School Attendance Reporting
    h. Test Scoring
    i. Student Programming
    j. Salary Schedules
    k. Accounts Receivable
    l. Gaming and Simulation

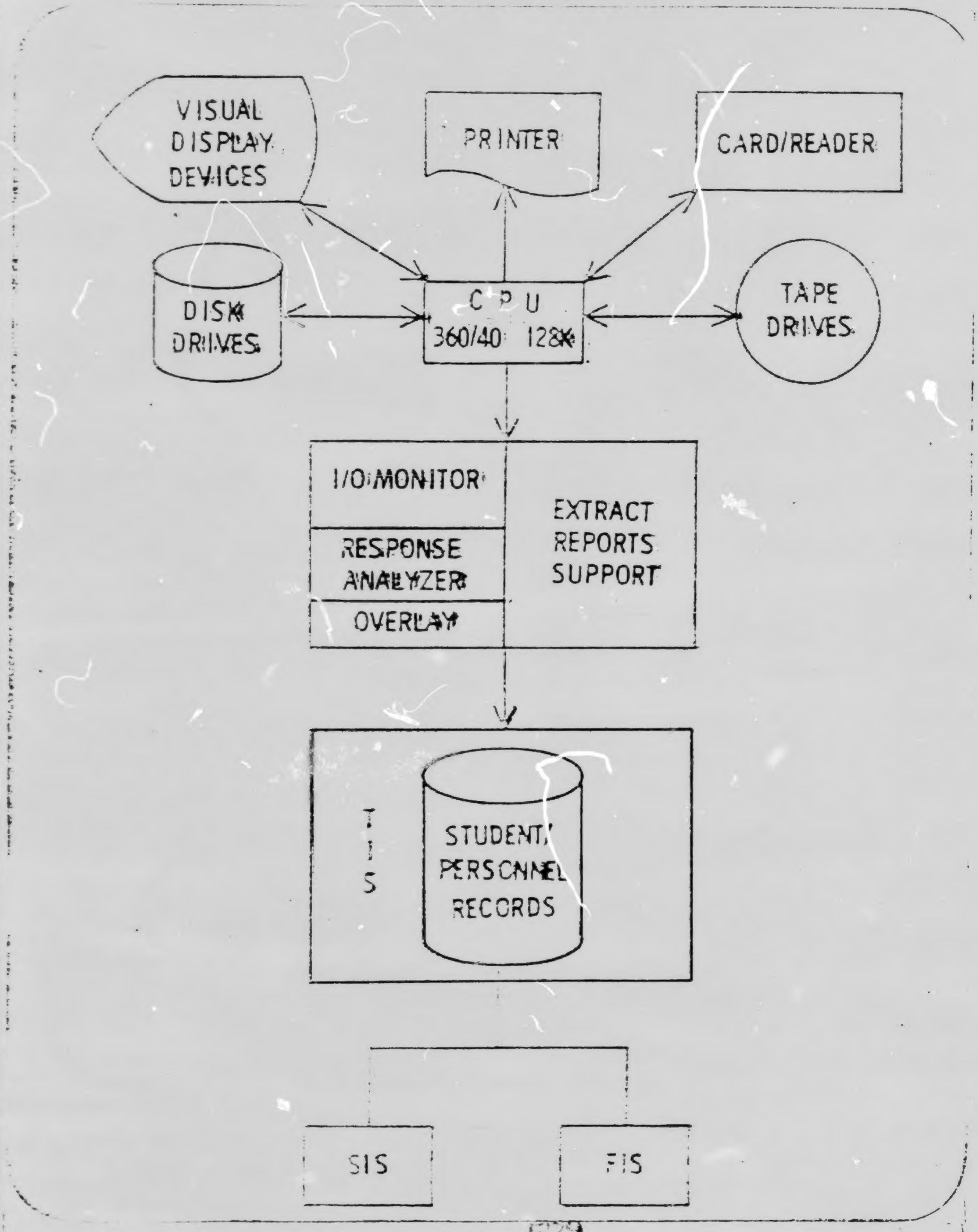
Profile of Users

Users of Data Processing services at College of DuPage are divided into two areas.

1. Internal users - Users within the College of DuPage are utilizing services of data processing in all areas of application. As of the date of this publication, the number of users are given below.
    - a. Test Scoring - 36 instructors
    - b. Gaming and Simulation - 2 instructors
    - c. Grade Reporting System - 390 instructors (full and part-time)
    - d. Administrative Reporting - 25 administrators
    - e. On-line Registration - all students (7833)
    - f. Drop/Adds - 3600 students
    - g. Advisor Reports - 275 instructors and administrators
    - h. Payroll - all college employees (624)
    - i. Student Programming - 300 students
    - j. Budget - 60 administrators and Board members
    - k. LRC Circulation - 22,143 books/quarter
    - l. Quarterly Schedule Publication - no estimate
2. External Users - Users of the facilities of the Data Processing Department outside of the immediate college community include 5 high schools, 11 junior high schools, 10 elementary school districts, 1 four-year college, and the DuPage County Health Department. In addition, negotiations are in progress for providing service for several municipal governments. A summary of users by application area is given below.
    - a. Scheduling - 19 schools
    - b. Payroll - 13 schools
    - c. CVIS - 5 schools
    - d. Grade Reporting Jr. High - 4 schools
    - e. Attendance - 4 schools
    - f. Grade Reporting Sr. High - 3 schools
    - g. Salary Schedule - 3 schools
    - h. On-line Registration - 1 school
    - i. Mailing Labels - 9 schools
    - j. Computer Assisted Instruction - 1 school
    - k. Budget - 2 schools
    - l. Computer Managed Instruction - 1 school
    - m. Test Scoring - 3 schools

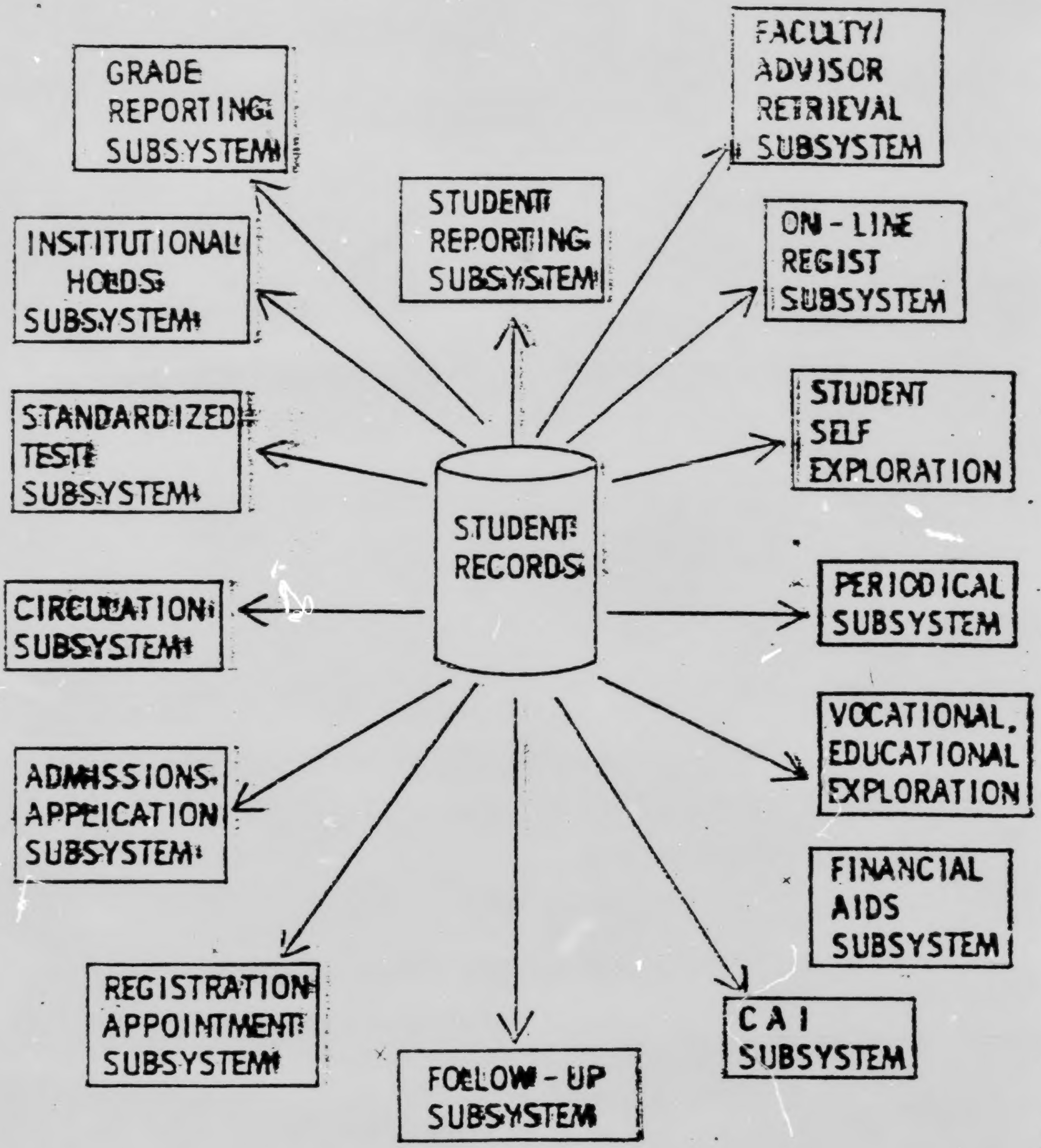
APPLICATION SYSTEM FLOWCHART

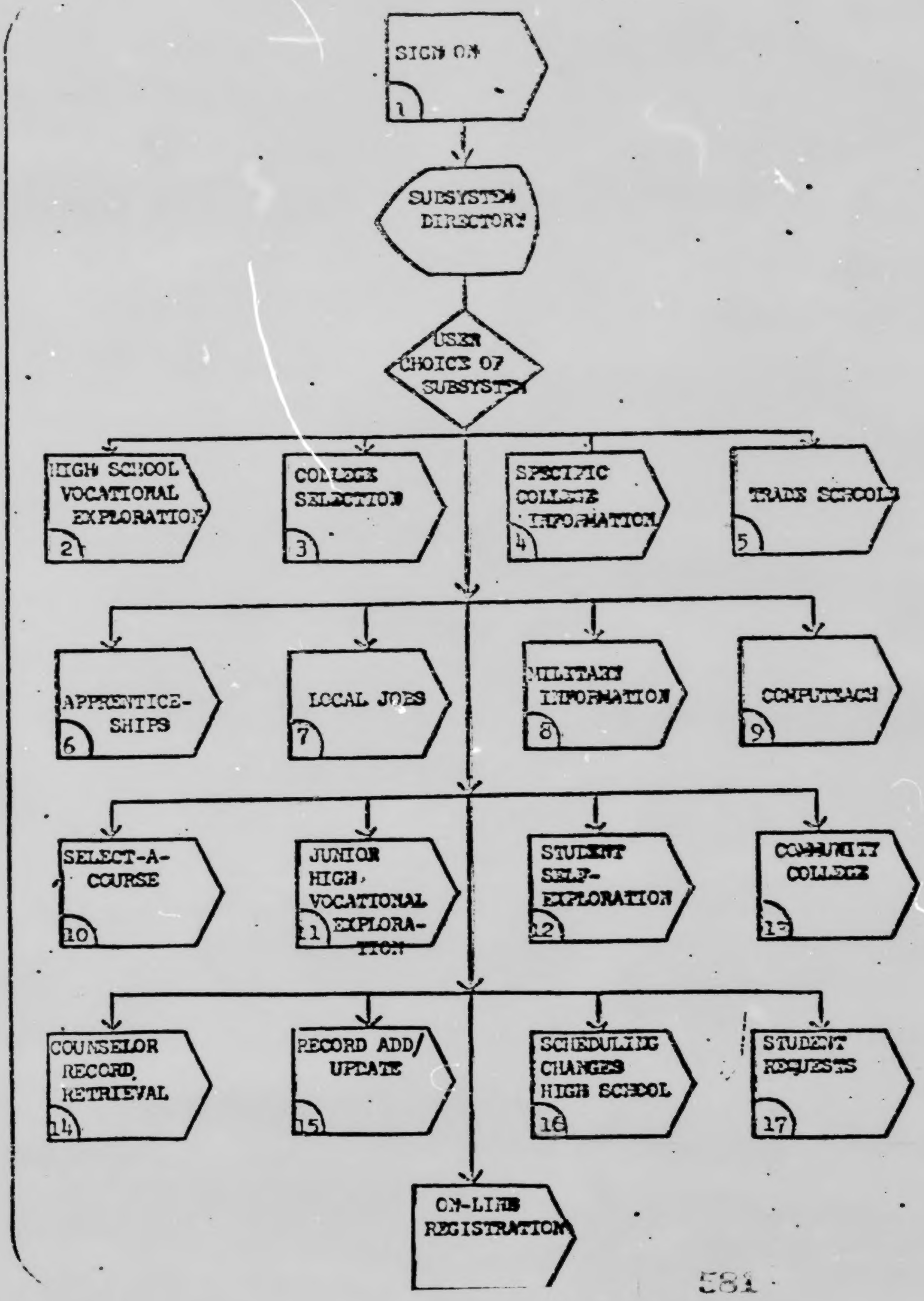

REPORTING
FILE MAINTENANCE
BATCH SUPPORT PROGRAMS
SYSTEM FILES
STUDENT
PERSONNEL
SCHEDULE
CIRCULATION
VENDOR
BUDGET
TEXT
MISC
ON-LINE APPLICATIONS
ADMINISTRATIVE
BUDGET
REGISTRATION
CIRCULATION
CLASS
CVIS
EXPORATION
TEST INTER.


VISUAL DISPLAY DEVICES
PRINTER
CARD/READER
DISK DRIVES
CPU
360/40 128K
TAPE DRIVES
I/O MONITOR
RESPONSE ANALYZER
OVERLAY
EXTRACT REPORTS SUPPORT
TIS
STUDENT/ PERSONNEL RECORDS
SIS
FIS

COLLEGE OF DUPAGE

STUDENT INFORMATION SYSTEM


STUDENT RECORDS
GRADE REPORTING SUBSYSTEM
STUDENT REPORTING SUBSYSTEM
FACULTY/ ADVISOR RETRIEVAL SUBSYSTEM
INSTITUTIONAL HOLDS SUBSYSTEM
ON - LINE REGIST SUBSYSTEM
STANDARDIZED TESTS SUBSYSTEM
STUDENT SELF EXPLORATION
CIRCULATION SUBSYSTEM
PERIODICAL SUBSYSTEM
ADMISSIONS APPLICATION SUBSYSTEM
VOCATIONAL, EDUCATIONAL EXPLORATION
FINANCIAL AIDS SUBSYSTEM
REGISTRATION APPOINTMENT SUBSYSTEM
FOLLOW - UP SUBSYSTEM
C A I SUBSYSTEM


SIGN ON
1
SUBSYSTEM DIRECTORY
USER CHOICE OF SUBSYSTEM
HIGH SCHOOL VOCATIONAL EXPLORATION
2
COLLEGE SELECTION
3
SPECIFIC COLLEGE INFORMATION
4
TRADE SCHOOLS
5
APPRENTICESHIPS
6
LOCAL JOBS
7
MILITARY INFORMATION
8
COMPUTEACH
9
SELECT-A-COURSE
10
JUNIOR HIGH VOCATIONAL EXPLORATION
11
STUDENT SELF-EXPLORATION
12
COMMUNITY COLLEGE
13
COUNSELOR RECORD RETRIEVAL
14
RECORD ADD/UPDATE
15
SCHEDULING CHANGES HIGH SCHOOL
16
STUDENT REQUESTS
17
ON-LINE REGISTRATION

GENERAL SYSTEM DESIGN

CVIS is a generalized on-line system for educational uses. It is designed to perform numerous functions for students, teachers, counselors and school administrators. It is composed of the following seventeen (17) subsystems and related functions which taken together identify the functional design of CVIS.

A. SYSTEM SUBSYSTEMS --

1. HIGH SCHOOL VOCATIONAL EXPLORATION - The subsystem enables the student to explore, in conversational mode, over 450 job descriptions contained in the system. Based on the student's choice of interest and educational plans, suggested job lists are presented. Student's decisions are related to the student's high school achievement and scores on standardized tests.
2. COLLEGE SELECTION - This subsystem is composed of ten (10) scripts related to choosing a college such as 'College Costs', 'How to Visit a College', etc., and logic modules required to enable a student to select a list of colleges. The student makes decisions on major, size, location, type of community, type of control, admission policies and cost. The program uses these decisions to select for the student a list of colleges from the 1300 in the system.
3. SPECIFIC COLLEGE INFORMATION - This subsystem enables students or counselors to retrieve information related to any of the 1300 colleges and universities in the system. Three types of displays are available to the student:

ENTRANCE REQUIREMENTS

MAJORS OFFERED

SUMMARY INFORMATION (Size, Cost, Location, etc.)

4. TRADE/TECHNICAL SCHOOL - This subsystem enables the student to explore information related to special trade and technical schools related to his choice of acception or interest area.

5. APPRENTICESHIP - This subsystem uses CAI approach to familiarize the student with general apprenticeship programs and then provides access to specific information related to special apprenticeship programs.

6. LOCAL JOB - This subsystem uses a gaming approach to the process of decision making related to job selection and then provides the student with information related to local job opportunities.

7. MILITARY INFORMATION - This subsystem is basically a CAI approach to information related to the student's decision regarding military obligations.

8. COMPUTEACH - This subsystem involves the use of computer-assisted-instruction. It provides the teacher or instructor with the capability of developing CAI material which can then be presented to students through the CVIS program.

9. SELECT-A-COURSE - This subsystem enables a student to select his next year's courses and have them stored on his record. It provides the student with the opportunity of reviewing his school record and checks his selected courses for pre-requisites.

10. JUNIOR HIGH VOCATIONAL EXPLORATION - This subsystem helps a student to select categories of jobs which he may explore. Based on thirty (30) questions presented to the student, Dr. John Holland's work categories are tested for the student in the order that they relate to the student's self profile.

11. STUDENT SELF-EXPLORATION - The subsystem is designed to enable the student at the community college level to explore his academic record, and relate it to his vocational and educational plans.

12. COMMUNITY COLLEGE - This subsystem enables a student to explore various curricula which are offered at the community college level and to relate course offerings to his vocational interests.

13. COUNSELOR RECORD RETRIEVAL - The subsystem is composed of a series of functions designed to give the school counselor quick, meaningful access to all parts of the student's academic record. Grades, rank, test information, personal information may be retrieved and displayed on the visual display terminal.

14. RECORD ADD/UPDATE - This subsystem enables authorized school personnel to add or update records to the school data base. Tight security is built into this subsystem for maximum protection of school records.

15. SCHEDULING CHANGES-HIGH SCHOOL - This subsystem enables school counselors or other authorized personnel to quickly and accurately change a student's current schedule. The changed schedule is displayed and may be printed so that the student can be given a hard copy.

16. STUDENT REQUESTS - This subsystem enables counselors or other authorized personnel to save on the student's record his requests for courses for the next school year. This is the counselor's version of the student subsystem 'SELECT-A-COURSE'. Course requests may be entered, added, deleted or changed. The saved requests become input to the IBM 360/scheduler program.

17. ON-LINE REGISTRATION - This subsystem is designed for registering college students. It provides on-line saving of requests, opening and closing sections, calculation of fees and printing of fees and schedules.

B. GENERAL SYSTEM DESCRIPTION

Below is a general description of CVIS which seeks to explain the design and programming concepts which enables a single program to perform the multi-functional goals indicated under the preceding heading.

1. OVERVIEW - CVIS is a 32K assembly language program which utilizes Basic Teleprocessing Access Methods (BTAM) macros to establish communication between 2260 visual display stations and Indexed Sequential Access Methods (ISAM) files. The controlling concept is that of providing communication between messages and other information contained on a disk file or files and the requesting display station. In order to accomplish the above objective, the CVIS MAINLINE program is divided into two main CSECTS which may be assembled separately and catalogued as a single phase program. The two CSECTS are

I/O MONITOR (CVISI0M)

RESPONSE ANALYZER (CVISRA)

2. I/O MONITOR - The I/O monitor has the basic responsibility of establishing and maintaining communication with all input/output devices in the CVTS network. More specifically it handles the following system functions.

    a) Opens all telecommunication lines and system files.

    b) Issues all read and write macros to local and remote display stations.

    c) Issues all write requests to 1053 printers in the network.

    d) Issues all read, update and add requests to system file.

    e) Monitors all telecommunication lines and stops activity on any line which develops trouble.

    f) Analyzes incoming and outgoing data for any loss [illegible] transmission and takes appropriate action

    g) Provides communication with the computer operator through the console typewriter.

    h) Identifies incoming requests with proper display station and routes outgoing messages to the proper destination.

    i) Translates incoming and outgoing data where necessary.

j) Passes to the RESPONSE ANALYZER all incoming data either from display stations or system files. Also properly identifies for the RESPONSE ANALYZER, the status of file handling requests.

k) Receives from the RESPONSE ANALYZER data to be displayed at visual displayed stations or data to add to or update records on file.

l) Provides timing factor to prevent writing to a busy 1053 printer.

m) Closes all communication lines and system files at shut down time.

n) Provides consistent displaying of data on both 960 character and 480 character screens.

o) Clears all input/output areas after handling of data.

3. RESPONSE ANALYZER - The response analyzer has the responsibility of handling requests which it receives from I/O MONITOR. It performs the following system functions:

a) Determines and controls usage of terminals and what functions may be accessed by terminals.

b) Checks requests from display station for codes which require special handling and provides the special handling.

c) Checks frames before displaying for any special functions to be performed.

d) Determines what frame is to be displayed next, requests retrieval of the frame by I/O monitor and then formats the frame for display.

e) Calls special functions from the text file, passes them to the overlay execution area and links to this area to execute functions or logic modules.

f) Communicates with I/O monitor to initiate print requests and formats data to be printed.

g) Checks student responses for correctness and selects proper feedback to student.

h) Intercepts any programming problems, logs these on program check log and notifies requesting station of the problem.

4. EXECUTION OF SPECIAL FUNCTIONS AND LOGIC MODULES - One of the unique features of CVIS is the manner in which special programming requirements are handled. All special functions and logic modules are programmed in basic assembly language, translated into machine code and placed as a record on the same file containing all system text data. These records are then accessed by their unique keys or codes, transferred to the overlay area in the RESPONSE ANALYZER and executed. This process is explained in detail in the section on "WRITING NEW APPLICATIONS".

1 - 2 2701 Transmission Control Units

1 - 31 2260 Visual Display Stations

1 - 2540 Card Reader/Punch

1 - 1403 Printer

1 - 2841 Control Unit

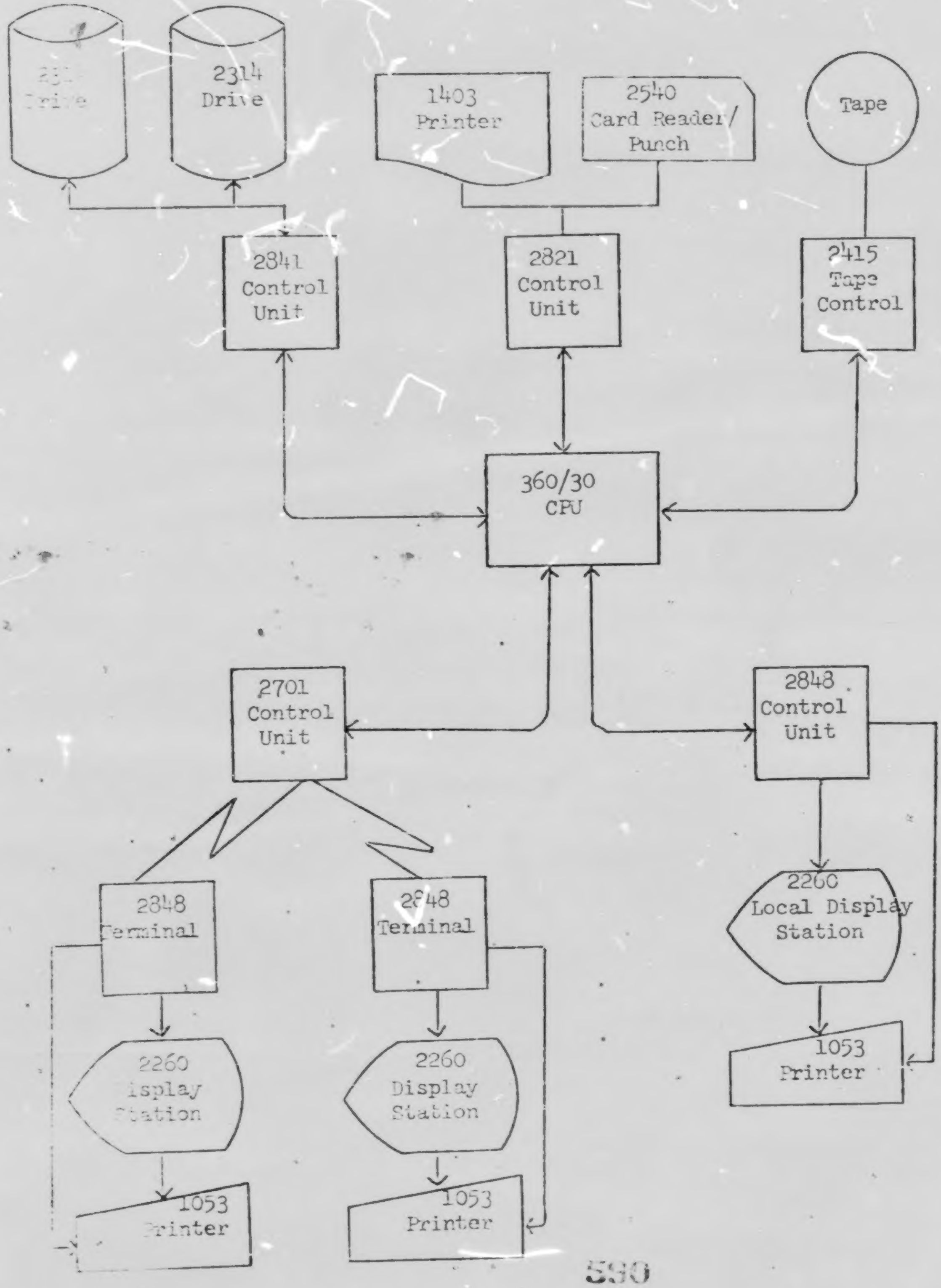
1 - 2821 Control Unit

1 - 2415 Tape Control Unit

BTAM Macros Catalogued in System Library

NOTE: While the above configuration calls for 2314 disk drives, the program is easily modified to 2311 drives. For details for this modification, see the section on "Program Modifications".

IBM 360/30 SYSTEM FLOWCHART


2314 Drive
2314 Drive
1403 Printer
2540 Card Reader/ Punch
Tape
2841 Control Unit
2821 Control Unit
2415 Tape Control
360/30 CPU
2701 Control Unit
2848 Control Unit
2848 Terminal
2848 Terminal
2260 Local Display Station
2260 Display Station
2260 Display Station
1053 Printer
1053 Printer
1053 Printer

5. LINKAGE BETWEEN FRAMES - Another unique feature of CVIS is the manner in which linkage is established between frames to be displayed at a visual display station. When a record is retrieved for displaying, the first forty (40) characters of the record contain control information which is used by the RESPONSE ANALYZER to establish linkage to the next frame. Frames are classified into certain basic types indicated by the fifth byte of the record. The RESPONSE ANALYZER uses this byte to determine how the frame is to be handled and what type of response is expected to the frame. The forty-byte control field is stored in the terminal table and saved until the next request is received from the terminal. The stored control field is then used to analyze the response from the terminal and to determine what action should be taken. The control field and its use is fully explained in the section on file description.

C. SYSTEM CONFIGURATION

CVIS operates in a multiprogramming environment using an IBM 360/30 65K or 360/40 central processing unit. The operating system is DOS Version 21. The specific hardware configuration depends on the installation under consideration. The following is a minimum system configuration:

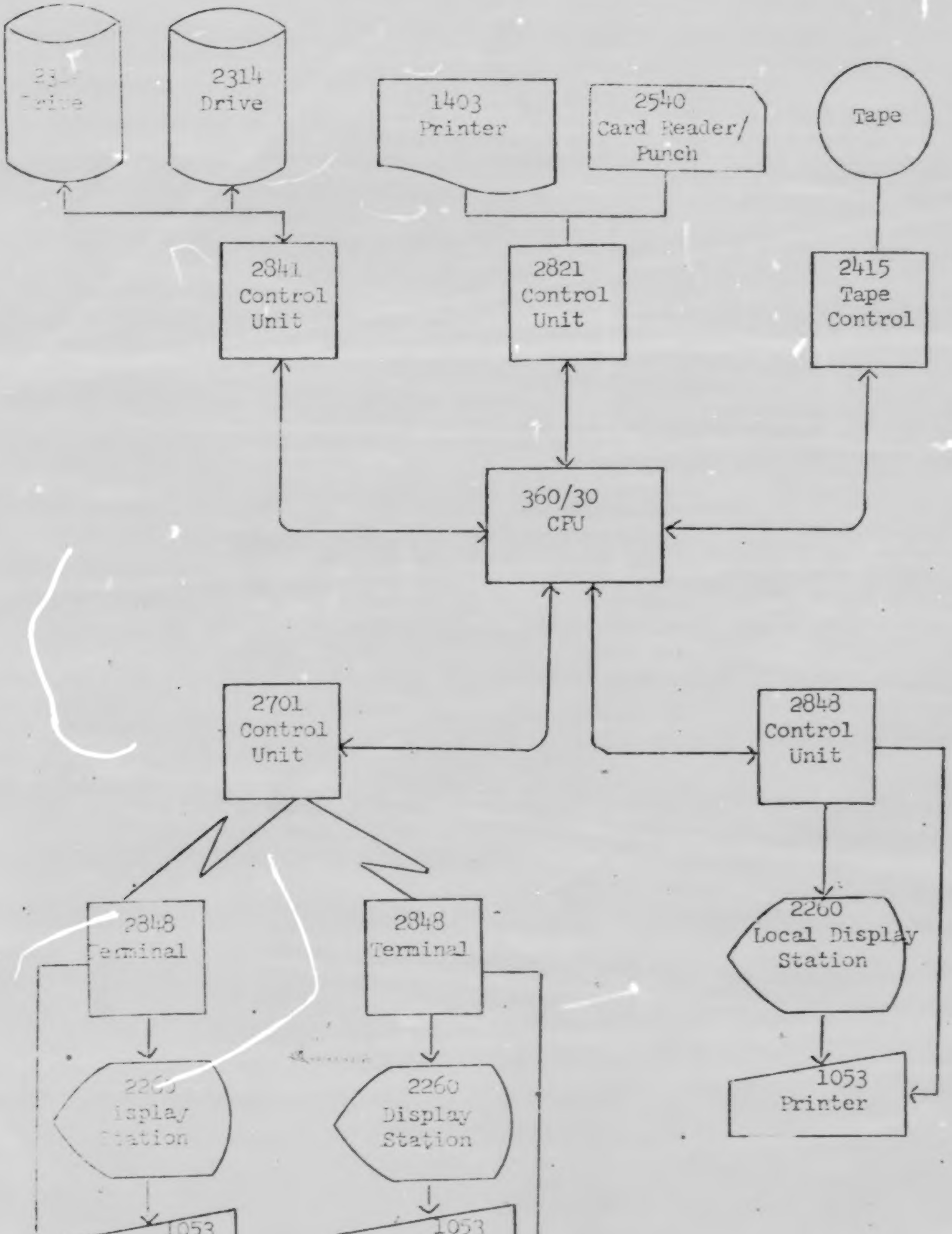
IBM 360/30 65K CPU

DOS Version 21

2 - 2314 Disk Drives

1 - 5.2848 Display Control Units

IBM 360/30 SYSTEM FLOWCHART


2314
Drive
2314
Drive
1403
Printer
2540
Card Reader/
Punch
Tape
2841
Control
Unit
2821
Control
Unit
2415
Tape
Control
360/30
CPU
2701
Control
Unit
2848
Control
Unit
2848
Terminal
2848
Terminal
2260
Local Display
Station
2260
Display
Station
2260
Display
Station
1053
Printer
1053
Printer
1053
Printer

E. GENERAL SYSTEM FLOWCHART

CVIS provides user access to three (3) different physical files used in the system. The three files are:

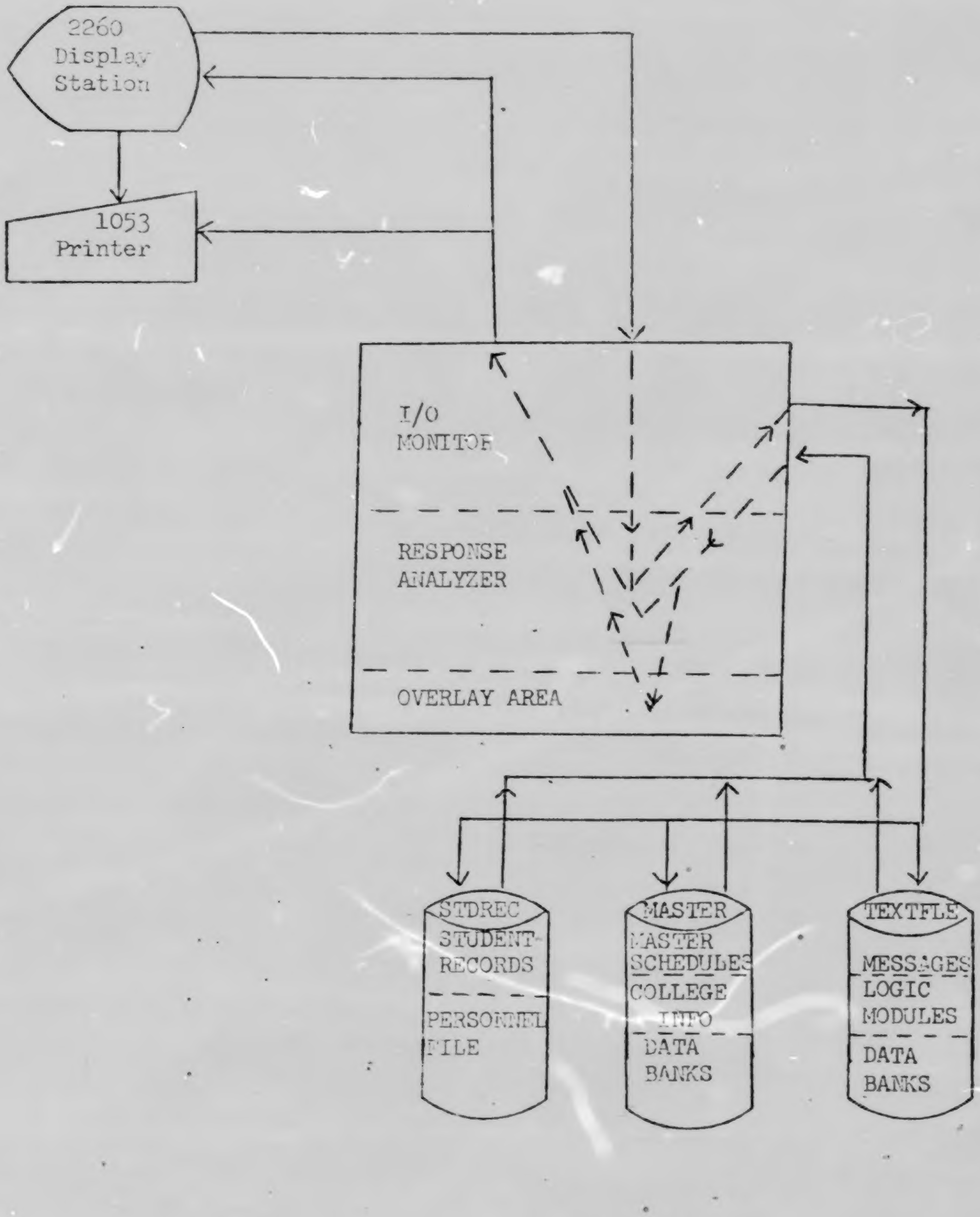
TEXTFLE - Containing all messages, logic modules and special data banks.

MASTERER - Containing school master schedules, college data banks and other special data banks.

STDREC - Containing Student and Faculty records.

On the following page is an overall diagram of the system.

CVIS SYSTEM FLOWCHART


2260
Display
Station
1053
Printer
I/O
MONITOR
RESPONSE
ANALYZER
OVERLAY AREA
STDREC
STUDENT
RECORDS
PERSONNEL
FILE
MASTER
MASTER
SCHEDULES
COLLEGE
INFO
DATA
BANKS
TEXTFLE
MESSAGES
LOGIC
MODULES
DATA
BANKS

2. EXPLANATION OF TABLES AND SPECIAL FLAGS

The following BTAM macros generate tables which are used to pass data to the BTMOD for execution of line control macros. The BTAM manual should be consulted for a detailed description of these tables.

DTF BT	(Define the file for BTAM)
READ DECB1, MF=L	(Generates DATA EVEN CONTROL BLOCK)
ASMTRTAB	(Translation Tables)
DFTRMLST	(Terminal Lists)

The following tables and flags provide data for use in I/O MONITOR and RESPONSE ANALYZER.

TERMTAB -- (Terminal Table)

This is a table containing 80 bytes for each terminal in the network. This table is used to store all data essential to unique communication with terminals. The address of the first entry for each line is identified by a label and the address of this label is stored immediately following the DECB. This address is used to aid in identifying the terminal being served. The table contains the following information for each terminal.

TERMINAL TABLE

POSITION	LENGTH	DESCRIPTION
1	1	Acceptable Terminal Use (ATU)
2	1	Current Terminal Use (CTU)
3	2	Terminal Address (TA)
5	4	Re-entry Address in RESPONSE Analyzer
9	2	Printer Address

TERMINAL ADDRESS -- This is a two-byte address of the display station. For remote terminals, it is two hex numbers, the first the address of the 2848 control unit and the second the address of the 2260. This address is compared with the terminal address in the DECB to identify the requesting display station. For local terminals, this address is a two-byte hex number. The first is a hex '43' (to distinguish it from remote terminals.) The second is the relative line number of the terminal. (From X'00' to X'0F' depending on the number of display stations).

RE-ENTRY ADDRESS -- This is a four-byte field for saving the re-entry address in the RESPONSE ANALYZER. This address is initialized to the INITIAL ENTRY ROUTINE in RESPONSE ANALYZER. It is modified by the RESPONSE ANALYZER so that it contains the address for returning to the appropriate routine in RESPONSE ANALYZER.

PRINTER ADDRESS -- This is a two-byte address of the 1053 printer assigned to the display station. The printer address must be the highest address of the devices attached to the 2848 control unit. For example, if four display stations are attached to a remote 2848, then the printer address would be X'40A4'.

SCHOOL CODE -- This is a two-byte code to identify the various schools served by the system. These codes must be determined prior to system implementation and must be the first two bytes of all file records unique to the school. For example, all student records on the student record file must have the school code in the first two bytes of the twelve-byte key. The school master schedule must have the school code as the first two bytes of the seven (7) byte key.

The school code is essential to the security provided by the system. In retrieving student records, the school code is supplied by the program from the terminal table. This prevents records being processed which do not belong to the school of the requesting display station.

STUDENT NUMBER -- This is a nine (9) byte field where the student number is saved. This allows repeated processing of a student record without the number having to be re-entered by the display station for each request. This field is replaced when a new number is entered.

RECORD NUMBER -- This identifies a record number on the student record file. A single student may be assigned multiple records. The records are numbers A, B, C, D, etc.

RESPONSE FIELD SAVE AREA -- This is a forty-byte area for saving the linkage information about frames. The RESPONSE ANALYZER stores the response field of frames in this area.

TERMINAL WORK/SAVE AREA -- This is an eighteen-byte area for temporarily storing data between displays on the display station. It is used for setting switches, saving counters, etc.

DECB EXTENSION -- Immediately following the Data Event Control Block for each line in the system is an eight-byte field used in the following way.

DCBFLAG - 1 byte -

Bit 0   1 = dial   0 = Leased Line or Local 2260

Bit 1   1 = Local 2260

Bit 2   1 = Remote 2260

Bits 1 & 2   If both 1, Remote 2265

Bit 3   1 = 1050 Not Currently Used

Bit 4 1 = 2740 Not Currently Used

Bit 5 1 = Line Open and Readable

Bit 6 1 = Line Offline, DTFBT not Opened

Bit 7 Not Used

TERMTAB ADDRESS -- 3 bytes

Address of first entry in the terminal table for this line.

PROCESS ROUTINE ADDRESS -- 4 bytes

Routine address to I/O Monitor to which control should be passed following a completed line I/O Operation.

ECBLIST -- See BTAM manual for explanation of this table. Also, see section on 'Program Modification' to see how this table must be modified when implementing CVIS.

PRINTBL -- (Print Table) This table contains 20 bytes for each 1053 printer in the network. The entries in the table are as follows:

PRINTER BUSY FLAG - 1 Byte

X'00' means printer available

X'01' means printer busy

This byte is set, checked and reset by the pointer routine in I/O MONITOR and RESPONSE ANALYZER.

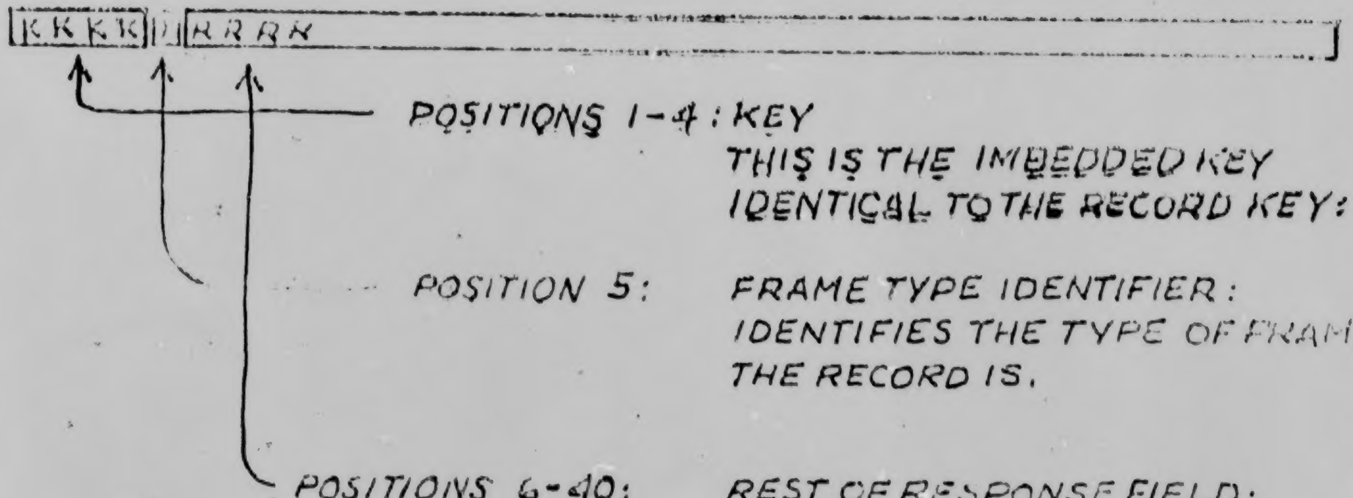
PRINTER ADDRESS -- 1 byte

address of the printer assigned to this table entry.

FILLER - 2 bytes -- Not used.

POLLING POINTER -- 4 bytes

Save area for polling pointer.



POSITIONS 1-4 : KEY
THIS IS THE IMBEDDED KEY
IDENTICAL TO THE RECORD KEY:

POSITION 5: FRAME TYPE IDENTIFIER:
IDENTIFIES THE TYPE OF FRAME
THE RECORD IS.

POSITIONS 6-40: REST OF RESPONSE FIELD:
USE OF THIS SUBFIELD IS
DEPENDENT UPON FRAME
TYPE.

TERMTAB ADDRESS -- 4 bytes

Save area for terminal table address of display station requesting printer.

DECB ADDRESS -- 4 bytes

Save area for DECB address of communication line.

TIME -- 4 bytes

Save area for storing time. This is used for timing routine to set or reset PRINTER BUSY FLAG.

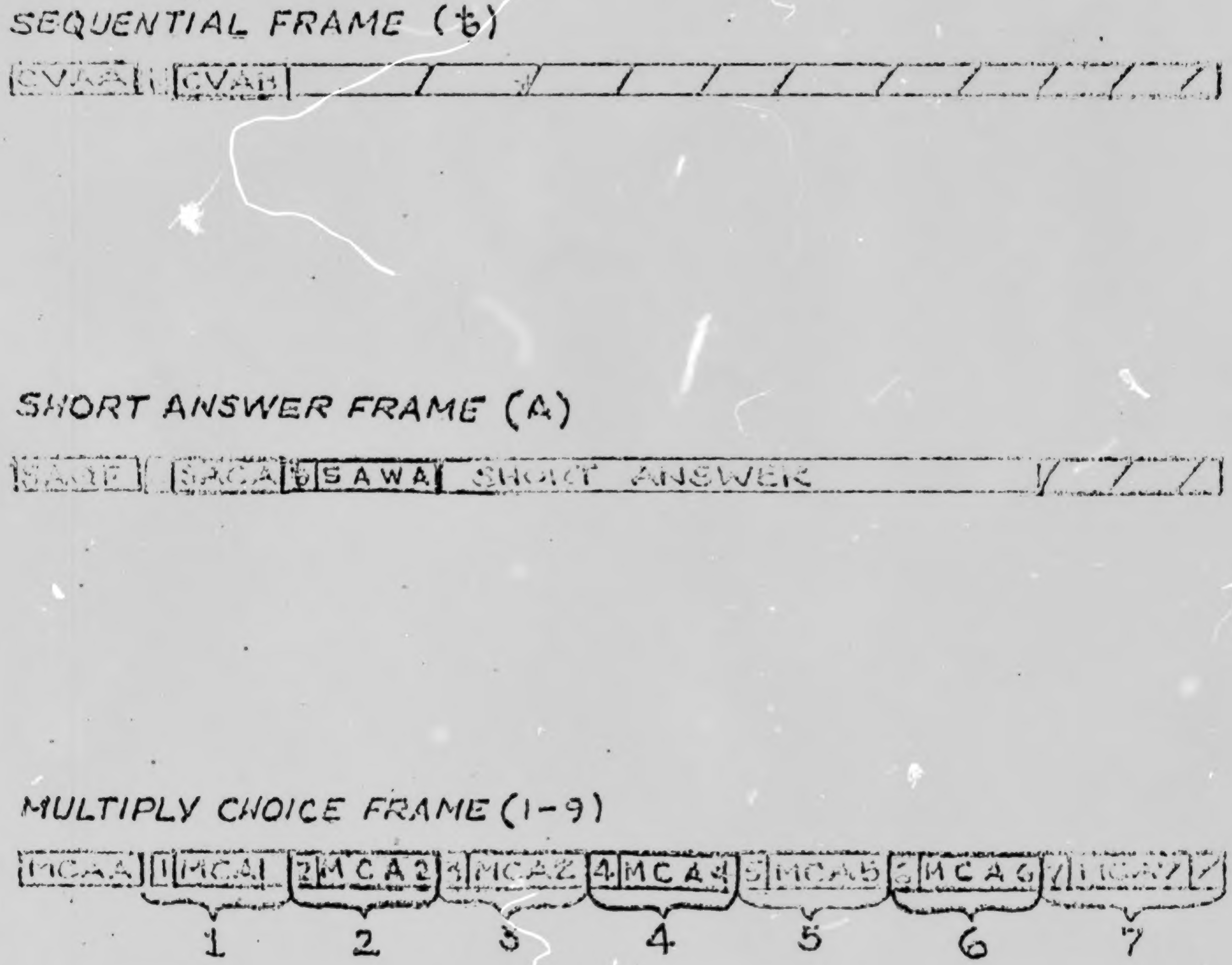
UPSIBYTE -- This is a one-byte field to save UPSI byte if UPSI card is read at Load Time. Bit on, associated DTFBT is not opened. Bits are

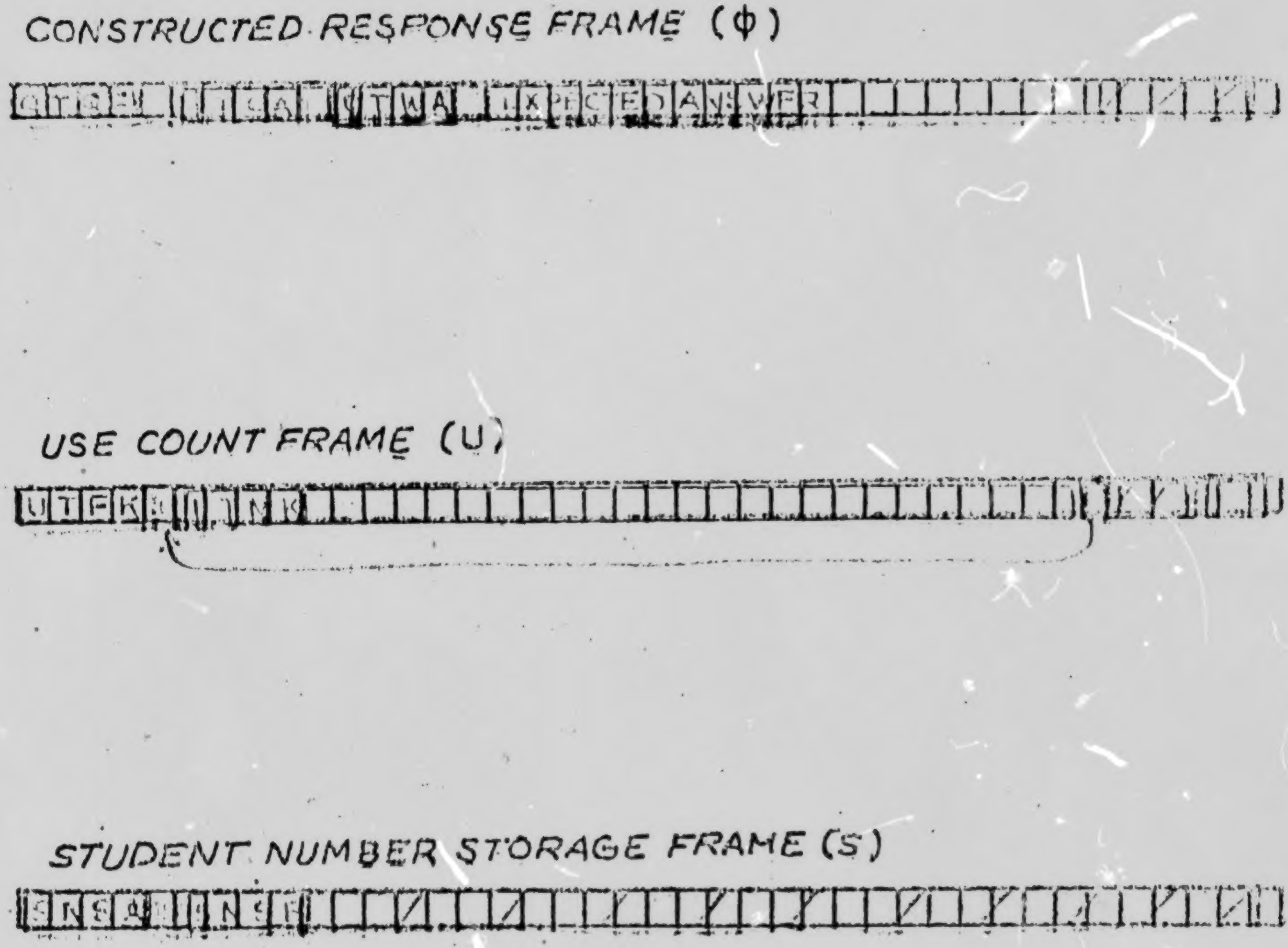
Bit 0 - Local DTFBT
Bit 1 - Remote DTFBT
Bits 2-7 Not Used

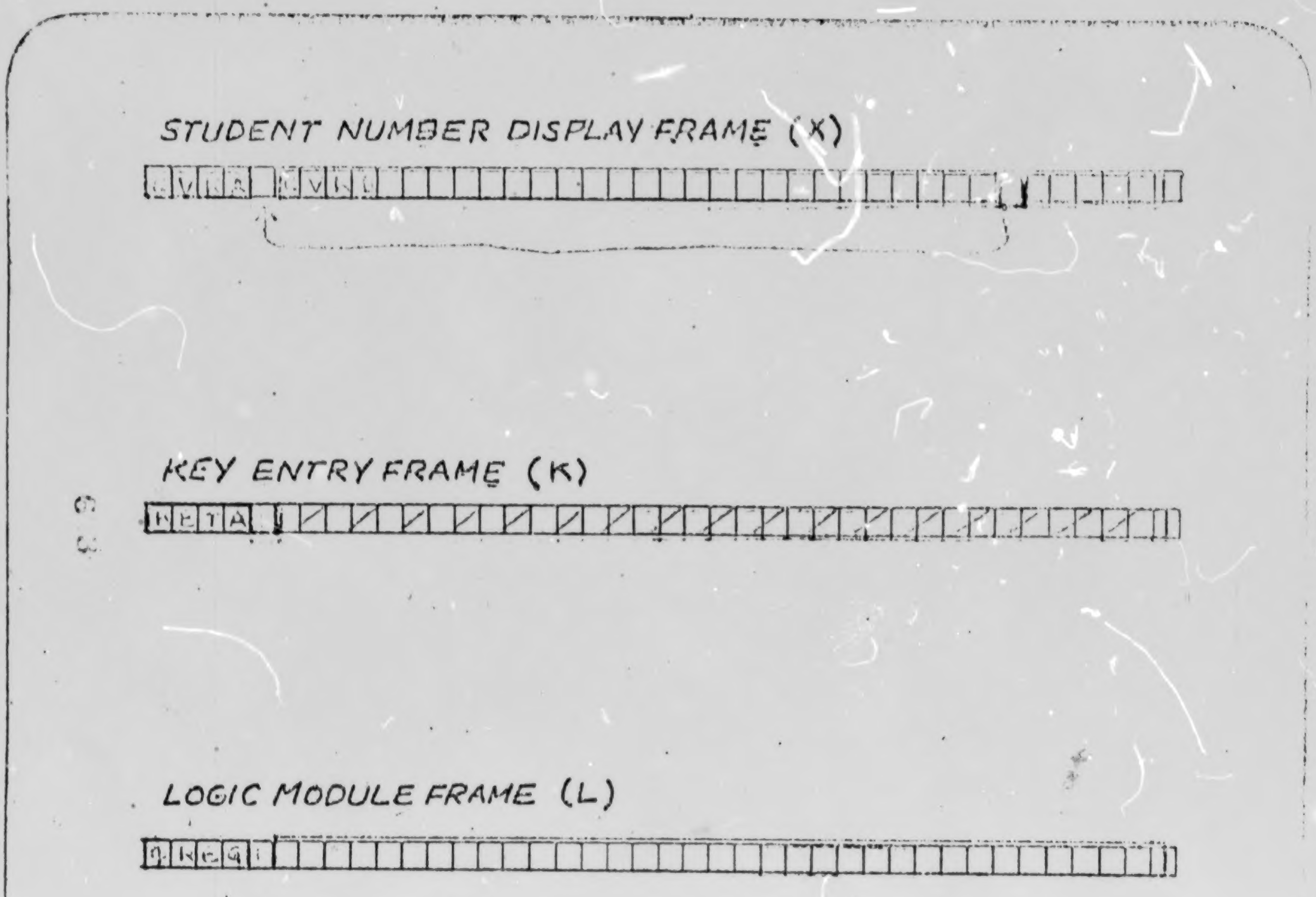
In addition to the above tables and flags, I/O MONITOR has an input/output area for each line in the system. For local terminals, there is one 960 byte area labelled 'LOCALIN'. Each remote line has a 484 byte area. For example, the current program has the following:

```
REMOTINW    DS  121F
REMOTINN    DS  121F
REMOTINY    DS  121F
```

There is also a 964 byte area for one remote 2265 labelled REMOTING.


SEQUENTIAL FRAME (b)
CVAA
CVAB
SHORT ANSWER FRAME (A)
SAQF
SACA
SAWA
SHORT ANSWER
MULTIPLY CHOICE FRAME (1-9)
MCAA
1
MCA1
2
MCA2
3
MCA3
4
MCA4
5
MCA5
6
MCA6
7
MCA7
1
2
3
4
5
6
7


CONSTRUCTED RESPONSE FRAME (Φ)
EXPECTED ANSWER
USE COUNT FRAME (U)
UTEK
STUDENT NUMBER STORAGE FRAME (S)
SNSA


STUDENT NUMBER DISPLAY FRAME (X)
KEY ENTRY FRAME (K)
LOGIC MODULE FRAME (L)

TOWARDS A NATIONAL POSTSECONDARY EDUCATION DATA BASE: EXPERIENCES OF THE NATIONAL COMMISSION ON THE FINANCING OF POSTSECONDARY EDUCATION

Daryl E. Carlson
James Farmer
Richard E. Stanton

PREFACE

In addition to the basic conclusions and recommendations of the National Commission on the Financing of Postsecondary Education, reported in Financing Postsecondary Education in the United States (Government Printing Office, December 1973), the staff prepared supplementary materials that were used by Commissioners in their deliberations. This staff report is one of a series by the staff to make available these materials to a broader audience. And, although these reports do not necessarily reflect the views or recommendations of the Commission, it is the Commission's privilege to publish them as a contribution to the current vigorous dialog on the financing of postsecondary education.

The National Commission on the Financing of Postsecondary Education (NCFPE) assembled a large data base of information about students and institutions and used a number of data processing techniques to support its analytical effort. These techniques included the use of on-line programming, statistical analysis (software) packages, interactive models, and a data manipulation and retrieval system. The principal problems in developing this data analysis capability were not technical. Rather, they were the economic and organizational problems inherent in obtaining machine readable data, resolving the definitional and coding inconsistencies, and training analysts to use new tools.

This staff report describes the content of the data base and the development and use of these techniques. This paper was presented to the College and University Systems Exchange (CAUSE) at a national convention in New Orleans on December 17, 1973.

George B. Weathersby
Director of Research

ACKNOWLEDGEMENTS

The data processing effort described in this report is the result of the special efforts of many different people. The basic data management language software—DS/3—is the result of the creative minds of Sally Bowman and Nancy Curtis at System Development Corporation, who provided assistance to the NCFPE staff. The data processing services required the patience and perseverance of Nick Corritori, Director of SDC's Corporate Computer Facility, and his staff; and Jim Jenkins and his colleagues at the Department of Health, Education, and Welfare's Data Management Center. These people accomplished many difficult tasks in unusually short periods of time and provided reliable data processing services throughout the project.

Many organizations assisted the Commission in obtaining the machine readable data to build its data base and this assistance is gratefully acknowledged. Sue Ellen Hais of the U.S. Office of Education provided assistance in obtaining and documenting the machine readable files from the Higher Education General Information Survey (HEGIS). Harold Nisselson of the National Center for Educational Statistics assisted in obtaining clearance for the Commission to receive HEGIS data. James Nelson and Ed Jacobsen of the College Entrance Examination Board assisted in obtaining data and getting permission to use student survey data from state agencies. Hayden Smith of the Council for Financial Aid to Education assisted in obtaining the Council's records on voluntary support of higher education.

We wish to extend to each of these people our appreciation for their support of our efforts and their willing acceptance of the often unrealistic demands imposed by the extremely short time available for the work of the National Commission.

LIST OF CHARTS

1 Data Processing Capabilities 12

2 National Postsecondary Education Data Base—File Directory 17

3 Record Structure of the Basic Traditional Institution Data Files 21

4 Record Structure of the Supplementary Traditional Institution Data Files 23

5 Sample DS/3 PRINT Command. 28

6 Sample DS/3 PRINT Dialog 29

7 Typical DS/3 Output, Major Research Universities . . . 30

8 Typical DS/3 PRINT Dialog with Calculated Values . . . 32

9 Probing a Data File with DS/3. 33

10 Illustration of SPSS Input Commands (Cross-Tabulations). 38

11 Illustration of SPSS Output (Cross-Tabulations). . . . 39

12 Illustration of SPSS Input Commands (Regression) . . . 40

13 Illustration of SPSS Output (Regression) 41

14 Terminal User Instructions from the Postsecondary Education Financing Model. 46

15 Sample of National Postsecondary Education Financing Model Dialog 47

16 Sample Output of the National Postsecondary Education Financing Model. 49

17 Comparative Data Services and Costs. 58

TABLE OF CONTENTS

		Page
	Preface	i
	Acknowledgements	iii
	List of Charts	v
I.	The National Commission and the Data Base Approach	1
II.	The Selection of Suitable Hardware and Software	7
III.	Building the Data Base	13
IV.	Retrieving Information from the Data Base	25
V.	Statistical Analyses	35
VI.	The National Postsecondary Education Financing Model	43
VII.	Users of the Data Base	51
VIII.	The Costs of Maintaining and Using the Data Base	55
IX.	The Future of the Data Base	59

I. THE NATIONAL COMMISSION AND THE DATA BASE APPROACH

The National Commission on the Financing of Postsecondary Education was a Congressionally authorized, Presidentially appointed seventeen-member body that began meeting in late 1972 to study the financing of U.S. postsecondary education.* Commission members were responsible for reporting to the President and Congress by December 31, 1973 the results of their study of:

- The impact of past, present, and anticipated private, local, state, and federal support for postsecondary education;
- The appropriate role for the states in support of higher education (including the application of state law on postsecondary educational opportunities);
- Alternative student assistance programs; and
- The potential federal, state, and private participation in such programs;
- The need, the desirability, the form, and the level of additional governmental and private assistance to postsecondary education;
- The nature and causes of serious financial distress facing institutions of postsecondary education; and
- Alternative models for the long-range solutions to the problems of financing postsecondary education with special attention to the potential federal, state, local, and private participation in such programs.

---

*The Commission was established by Public Law 92-318 (Section 140). Of the seventeen members, thirteen were appointed by the President, two were appointed by the President of the Senate, and two by the Speaker of the House. Its full report is found in Financing Postsecondary Education in the United States (Washington, D.C.: Government Printing Office, December 1973).

With only several months available for research, the Commission faced the task of developing a research strategy that would yield results quickly and creatively. In this regard, the Commission decided to emphasize an analytical approach drawing upon the wealth of data that had been collected by earlier commissions and study groups, but had not yet been analyzed fully.

The first part of the Commission's effort was spent in identifying appropriate objectives for postsecondary education, discussing trends and major issues, and studying the current financial situation. At the same time, the identification of applicable data, obtaining their use, and analyzing them in the context of an analytical framework was a staff responsibility. It became immediately evident that there were large quantities of data available in machine readable form, but that simple tabulations would not be satisfactory to describe and illuminate the complex and subtle processes of postsecondary education.

The strategy selected for handling all of these data was for the staff to identify the general types of data that would be desirable. Then, individual staff members were designated to obtain the data within their area of expertise. In a few weeks, such an approach could produce significant quantities of relevant, machine readable data. The Commission's work could not await the exercise of the more traditional approach of defining the problem, obtaining data, performing analysis, and then redefining the problem when the available turnaround for programming was days the analysts had no flexibility in specifying what they wanted.

The Commission staff chose a new approach which differed from the traditional approach in two important ways. First, organizing the data base was a separate task from analysis. The data base was expected to answer many yet-unformulated questions. Second, analysts interacted directly with the data base without programmers as intermediaries.

This new strategy met the Commission's special needs in analyzing current and proposed postsecondary financing plans. The data base and associated software also met these additional criteria:

- o Flexibility in structuring the data: To be of assistance to Commissioners, the staff had to access as complete and as unstructured a data base as possible. It had to be free to tabulate institutional data, for example, on different dimensions: by public and private control, by Carnegie Commission institutional classification, by size of enrollment, by geographic region, or by tuition level.
- o Rapid retrieval and query of the data: The research staff had to have the capacity for summarizing the data in a variety of ways: such as by totals and averages of different variables, (enrollment, finance, degrees) for different subsets of institutions; or frequency distributions of students for different variables (family income, race, sex, level). This kind of information had to be generated within hours or days, not weeks or months.
- o Statistical analysis: A statistical routine familiar to the staff members had to be selected, for there was little time to learn new statistical languages. As a result, SPSS (Statistical Package for Social Sciences) and BMD (Biomedical Statistical Programs) were chosen.
- o Special analyses: Ease of manipulating data in various ways was paramount. Fortran or PL/I were chosen to give the staff the capability of writing special purpose programs. In addition, for developing a model for analyzing the impact of financing plans on postsecondary education objectives, the Commission needed on-line program input as well as debugging and editing capabilities.
- o Non-technical staff: The software system for retrieving data was oriented towards users with limited computer programming skills.

- Hours of computing service: Since the staff had to be prepared for the Commission's monthly meetings it needed access to a reliable computing service for at least sixteen hours each day.

The staff estimated that developing a data base that met all of these requirements would require three to six months and $50,000. But only two months and $10,000 were available. After three weeks, the staff had the first institutional file on-line; and other data files were soon available.

II. THE SELECTION OF SUITABLE HARDWARE AND SOFTWARE

Five criteria were used by the staff in selecting a data inquiry or retrieval language capable of meeting the Commission's many needs:

- The computer language would have to be capable of selective retrieval, formatted reports, computed values, and elementary statistics.
- It should already be in use—preferably fully implemented, with a history of operating reliably at an available service bureau.
- It should be readily usable by analysts without extensive training.
- It should operate on a data base that could concurrently use Fortran, PL/I, and COBOL programs.
- It should be reasonably economic and, if possible, available within the federal government.

After some staff investigation, System Development Corporation's DS/3 language was selected. The language appeared to have several advantages for the Commission's use. First, it had the required language capabilities. Second, and quite important, DS/3 operated on standard IBM OS data sets. This capability meant that the development of Fortran models could proceed concurrently, and that standard statistical packages, like the Statistical Package for Social Sciences (SPSS), could be used.

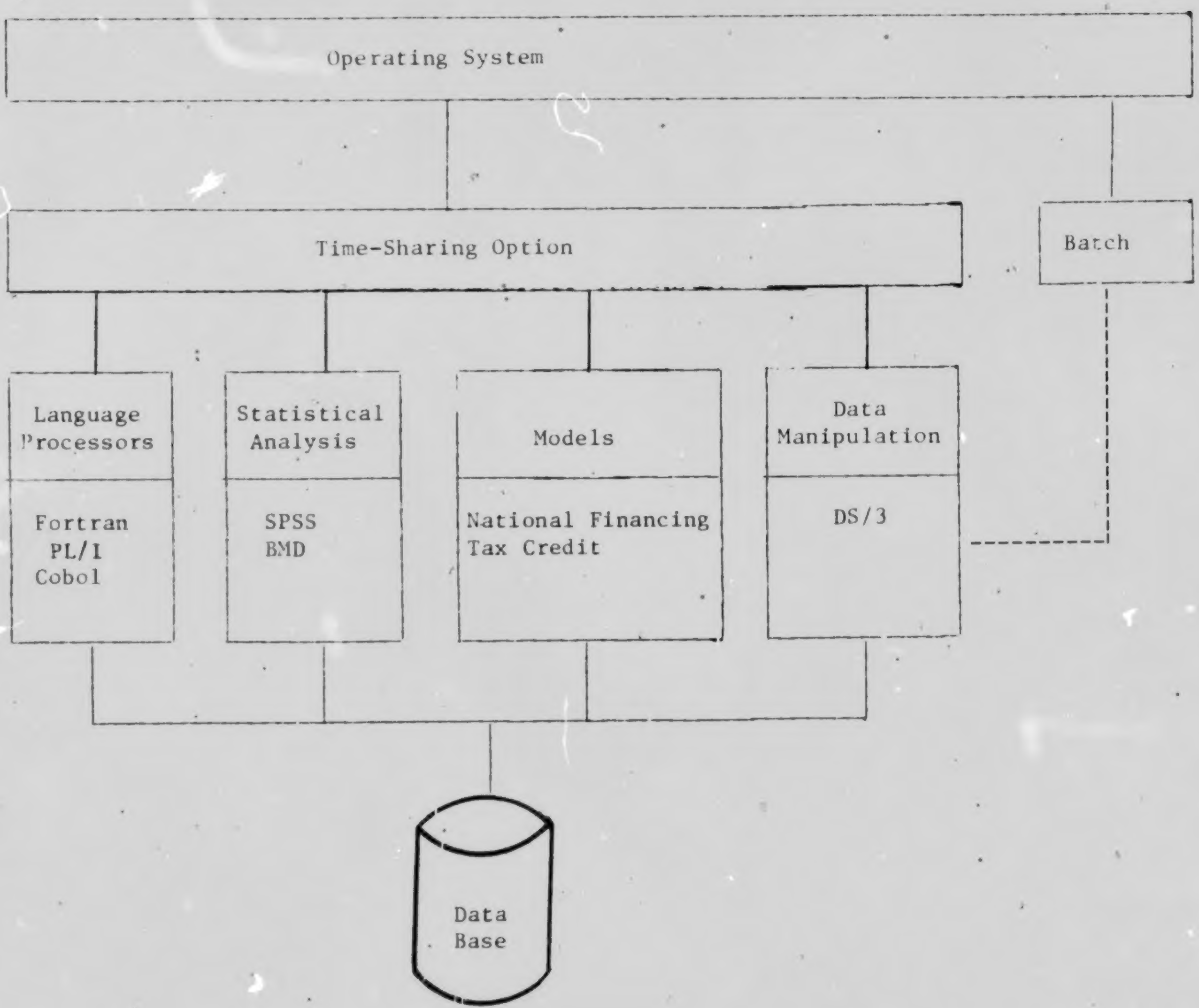
Third, DS/3 was expected to be operational in the federal government, specifically at the Data Management Center of the Department of Health, Education, and Welfare, by mid-August 1973. Also, the System Development Corporation (SDC) had had DS/3 fully operational at the Santa Monica Corporate Computer Facility for several months. Fourth, the documentation of the system was reasonably clear and the language seemed to be intuitively clear to non-programmers.

The staff also took care in selecting the most appropriate hardware for the Commission's computing needs. The System Development Corporation's Corporate Computer Facility provided the National Commission with several desirable features. First, the data base was implemented on an IBM 370/158 computer operating under VS2 at SDC. The characteristic of the virtual system (VS2) enabled SDC to provide responsive on-line computing service through time-sharing terminals to many users simultaneously. Second, SDC had a number of IBM 3330 disk storage units on-line that provided adequate storage space for the Commission's many data files (approximately 80 million bytes). Third, SDC's Corporate Computer Facility offered time-sharing services every day except Sunday from 5:00 a.m. to midnight Pacific time. Sundays, upon request, it was available from 1:00 to 9:00 p.m. Fourth, the SDC system rarely was inoperable.

In addition to the SDC facility, the National Commission also maintained two of the institutional data files at the Data Management Center of the Department of Health, Education, and Welfare in Washington, D. C. Because some of the institutional finance data was confidential, it was not possible to keep all of the finance data in the file at SDC in Santa Monica. On the other hand, the Time-Sharing Option (TSO) was not available at HEW's Data Management Center, so that special programs like SPSS and BMD could not be used at the D. C. facility. For this reason, as much of the data as possible was maintained in the Commission files at SDC in Santa Monica.

The relationship between the data base, the models, DS/3, the statistical packages, and the language processors is shown on Chart 1. Because of the characteristics of TSO and its associated software, all of these capabilities were available to the terminal user in conjunction with the NCFPE data base. This combination of data, models, and software represented a significant increase in the availability and usefulness of data on postsecondary education.

Chart 1: Data Processing Capabilities


Operating System
Time-Sharing Option
Batch
Language Processors
Fortran
PL/I
Cobol
Statistical Analysis
SPSS
BMD
Models
National Financing
Tax Credit
Data Manipulation
DS/3
Data Base

III. BUILDING THE DATA BASE

After the staff reviewed the types of data needed and the language and systems to be used, they designed an overall data base structure. An institutional file based on the Higher Education General Information Survey (HEGIS) of the National Center for Education Statistics formed the basis of data for collegiate institutions. Surveys of students and student aid recipients were added to the data base to provide information on student characteristics, tuition levels and other costs and student aid programs. Information on the financing programs of the federal and local governments was developed, based on information from the Office of Management and Budget (OMB) and the Census of state and local governments. In addition, the U. S. Office of Education's survey of noncollegiate schools and the Carnegie Commission's survey of proprietary institutions (private technical, business, specialized, and vocational schools and colleges) were incorporated. The data base was supplemented with the Council for Financial Aid to Education's data on voluntary support of colleges and universities.

Chart 2 lists the files that were available to the Commission. Some of these files were public; that is, anyone could use the file, and it could be duplicated and distributed without restriction. Some of the files were restricted by the source to use by Commission staff members; consequently, other users needed to obtain permission directly from the data source to gain access to such files.

All files were protected by security keys (passwords). There were two kinds of passwords, and each changed monthly. First, a password was needed to gain access to the data processing system. Second, each of the protected files had a security key, or password, of random letters. The password system presented two operational problems: (1) appropriate staff had to be assured of receiving the needed security keys; and (2) the keys, random letters, were difficult, if not impossible to remember.

617

The HEGIS surveys presented a special problem. In order to be readily usable by standard statistical packages and by the data retrieval language (DS/3), the results of several different surveys had to be combined into a single record. Chart 3 shows how the surveys were combined to form TRINST71, an institutional file for 1970-71. This file contained information collected between July 1970 and October 1971. As a result, the basic file contained 633 data elements and the record length was over 3,400 characters long. This length was satisfactory for use by both DS/3 and SPSS without significantly degrading performance. However, data on employees, basic student charges, and physical facilities were maintained in a separate, though linked, file to reduce retrieval costs.

The use of HEGIS data presented a serious policy issue about the legitimate use of restricted financial and employee data. Through the cooperation of the NCES staff, the issue was resolved. Salary data in the employee's file was excluded. And data from all institutions that had requested confidentiality were deleted. In addition, enough additional data from other institutions was dropped, called secondary suppression, to assure that restricted institutional data could not be discovered from combined use of data from the file and from previously published totals. These exclusions resulted in an "expurgated" finance file that was incorporated into the public HEGIS file. Of course, the Commission needed full finance data, so that special arrangements were made for selected staff members to have access to these data using the Department of Health, Education, and Welfare's Data Management Center facility. For the Commission's study on financial distress, HEGIS finance data for 1968-72 was provided through batch processing and NCES review. (See Chapter 5 of Financing Postsecondary Education in the United States.)

Chart 2

National Postsecondary Education Data Base
-File Directory-

	File	Contents	Status
(1)	TRINST71	Office of Education, HEGIS Surveys, 1970-71 1) Institutional characteristics 2) Opening fall enrollment 3) Finance 4) Earned degrees (637 variables, 3265 institutions)	Public
(2)	TRINST72	Office of Education, HEGIS Surveys, 1971-72 1) Institutional characteristics 2) Opening fall enrollment 3) Finance (278 variables, 3407 institutions)	Public
(3)	TRINST73	Office of Education, HEGIS Surveys, 1972-73 1) Institutional characteristics 2) Opening fall enrollment (173 variables)	Public
(4)	VOLSUP72	Council for Financial Aid to Education, Voluntary Support Survey, 1971-72 (private support to institutions by source and use) (156 variables, 1450 institutions)	Public
(5)	CEEBCLS	College Entrance Examination Board, College Locater Institutional File, 1972-73 (institutional characteristics, activities, programs; student characteristics) (139 variables, 2640 institutions)	Protected

File	Contents	Status
(6) TRINST71b	Office of Education, HEGIS Surveys, 1970-71 1) Institutional characteristics 2) Opening fall enrollment 3) Employee (no salary data) 4) Physical facilities 5) Basic student charges (3265 institutions)	Public (not yet available)
(7) TRINST72b	Office of Education, HEGIS Surveys, 1971-72 1) Institutional characteristics 2) Opening fall enrollment 3) Physical facilities 4) Basic student charges (3407 institutions)	Public (not yet available)
(8) TRINST73b	Office of Education, HEGIS Surveys, 1972-73 1) Institutional characteristics 2) Opening fall enrollment 3) Employee (no salary data) 4) Basic student charges	Public (not yet available)
(9) VOLSUP71	Council for Financial Aid to Education, Voluntary Support Survey, 1970-71 (private support to institutions by source and use) (156 variables)	
(10) CALVERT	Office of Education, Vocational Education Directory Survey, 1970-71 (347 variables, 11731 institutiona)	Public
(11) STSCHLR	State Scholarship Surveys of Financial Aid Recipients, 1971-72, for California, New York, New Jersey, and Pennsylvania) (64 variables, 3110 students)	Protected

	File	Contents	Status
(12)	SRSCAUN	CEEB, College Scholarship Service, Student Resource Surveys, 1973-74, University of California campuses. (69 variables, 63740 students)	Protected
(13)	SRSCASC	CEEB, College Scholarship Service, Student Resource Surveys, 1973-74, California state colleges and universities (69 variables, 47252 students)	Protected
(14)	SRSCAPC	CEEB, College Scholarship Service, Student Resource Surveys, 1973-74, California private four-year colleges and universities (69 variables, 12182 students)	Protected
(15)	SRSCATY	CEEB, College Scholarship Service, Student Resource Surveys, 1973-74, California private two-year colleges (69 variables, 37696 students)	Protected
(16)	SRSPENN	CEEB, College Scholarship Service, Student Resource Surveys, 1973-74, Pennsylvania public and private colleges and universities (69 variables, 19793 students)	Protected
(17)	SRSOREG	CEEB, College Scholarship Service, Student Resource Surveys, 1973-74, Oregon public and private colleges and universities (69 variables, 32248 students)	Protected
(18)	SRSWASH	CEEB, College Scholarship Service, Student Resource Surveys, 1973-74, Washington public and private colleges and universities (69 variables, 27624 students)	Protected

File	Contents	Status
(19) CARNEGIE	Carnegie Commission, Survey of Private, Technical, Business, Specialized and Vocational Schools and Colleges, 1972 (158 variables, 674 institutions)	Protected
(20) NCFPEVT	National Commission on the Financing of Postsecondary Education, Survey of Postsecondary Career Schools, 1973 (71 variables, 227 institutions)	Protected
(21) LOCAL	NCFPE and the Bureau of the Census, Local Government Funding data, 1966-67, 1971-72 (75 variables, 403 districts)	Protected
(22) FEDERAL	National Commission on the Financing of Postsecondary Education, Federal Program Funding data, 1971-74 (26 variables, 389 programs)	Protected
(23) ISSC	Illinois State Scholarship Commission, Survey of Financial Aid Recipients, 1970-71 (48 variables, 1294 students)	Protected

Chart 3: Record Structure of Basic Traditional Institution Data Files

TRINST71
(fiscal 1971)

	Institutional Characteristics	Opening Fall Enrollment	Finance	Earned Degrees
Variables:	1 - 74	75 - 172	173 - 291	292 - 633
Characters:	1 - 111	112 - 699	700 - 1770	1771 - 3484

TRINST72
(fiscal 1972)

	Institutional Characteristics	Opening Fall Enrollment	Finance
Variables:	1 - 74	75 - 172	173 - 291
Characters:	1 - 111	112 - 699	700 - 1770

TRINST73
(fiscal 1973)

	Institutional Characteristics	Opening Fall Enrollment
Variables:	1 - 74	57 - 172
Characters:	1 - 111	112 - 699

In addition to TRINST71, TRINST72, and TRINST73, many other files were created (see Chart 2). Three supplementary files of HEGIS data are being constructed in a manner similar to the basic TRINST files (see Chart 4). The remaining collegiate institutional files VOLSUP72 and CEEBCLS) are organized with one record per institution. Each record contains information about the institution's characteristics identical to that in the TRINST files. FICE codes for each institution are in all of these files, so that data from different files can be matched.*

Three files were constructed from data on noncollegiate or vocational and trade institutions (CALVERT, CARNEGIE, NCFPEVT). Again, FICE codes identifying institutions provide a common link between each of these files.

Nine different files of student data were developed from various surveys of students. Two of the surveys (ISSC and STSCHLR) were limited to financial aid recipients. For all of these files, each record represented the response of one student to the respective questionnaire.

The FEDERAL data file represents data on federal government postsecondary education financing programs with each record representing data on one specific program.

The local data file consists of data on revenue sources for local districts that manage their own in postsecondary educational institutions. Each record represents a specific district.

The combination of all these data files forms a data base of twenty-three files using over 110 million bytes of direct access storage. A fully documented description of all the data files is contained in a separate staff report of the National Commission, entitled "NCFPE National Postsecondary Education Data Base Directory."

---

*The Federal Interagency Committee on Education assigns unique codes for each institution.

Chart 4: Record Structure of the Supplementary Traditional Institutional Data File

TRINST71B (fiscal 1971)	Institutional Characteristics	Opening Fall Enrollment	Employees	Physical Facilities	Basic Student Changes
Variables:	1 - 74	75 - 172	173 -		
Characters:	1 - 111	112 - 699	700 -		

TRINST72B (fiscal 1972)	Institutional Characteristics	Opening Fall Enrollment	Physical Facilities	Basic Student Changes
Variables:	1 - 74	75 - 172	173 -	
Characters:	1 - 111	112 - 699	700 -	

TRINST73B (fiscal 1973)	Institutional Characteristics	Opening Fall Enrollment	Employees	Basic Student Changes
Variables:	1 - 74	75 - 172	173 -	
Characters:	1 - 111	112 - 699	700 -	

IV. RETRIEVING INFORMATION FROM THE DATA BASE

Both the number of data sets in the NCFPE data base (listed in Chart 2) and the size of many of the data sets are extensive. Of great service to the research staff was the capacity to access any piece of information in a matter of seconds. At the peak of staff research activity, all the data files together totaled over 120 million characters of data. Most of the data are as unstructured as possible; they are stored in basic, but edited form. The hardware and software that the staff selected allowed the analyst to access the data quickly and to structure the data to suit information needs. To indicate the data retrieval and query capabilities of the NCFPE data base, available through DS/3, a few examples are useful.

Chart 5 illustrates the basic DS/3 PRINT command. The date elements can be referred to by name, as shown, or by element number. The PRINT statement permits immediate selection by using WHERE (or WH). Chart 5 shows a typical retrieval of data for Carnegie Classification 11, major research universities. Headings are supplied automatically by DS/3. The user is given the opportunity to print a small sample, to search the entire file, or to process the inquiry in a batch mode. Chart 5 shows a sample; 'Y' signifies the user's response.

Chart 6 shows a more extended dialog where a sample (PROOF) of the report was requested and where sums and averages were added to the report. Since there are a number of systems where the institutional enrollment is zero, like the University of California system office, the user in this sample requested that the FICE code, institution name, Carnegie classification, AAUP faculty salary classification, total enrollment, and total graduate degree resident enrollment be printed only for the major research universities with enrollments greater than zero. Chart 7 gives the resulting output from the dialog on Chart 6. (Tick marks have been inserted here to signify the user's commands.)

Chart 5: Sample DS/3 PRINT Command

```
-PRINT FICE, INSTNAME, CARNCODE, AAUPCODE, ENTOT, ENFTE WHERE CARNCODE
-EQ 11

   ·68 COLUMNS-(Y/F/B):
-Y

 INST                              CARN    AAUP    FALL 71      FALL 71
 FICE                              CLASS   FACSAL   TOTAL       TOT FTE
 CODE                              CODE    CODE    ENROLLMENT   ENROLLMENT
  1131 CAL INST OF TECHNOLOGY       11       0       1,504        1,504
  1305 STANFORD UNIVERSITY          11       7      12,479       12,479
  1312 U OF CAL BERKELEY            11       0      28,384       28,384
  1313 U OF CAL DAVIS               11       0      14,571       14,571
  1315 U OF CAL LOS ANGELES         11       0      33,837       33,837
  1317 U OF CAL SAN DIEGO           11       0       7,037        7,037
·(Y/F/B):
```

Chart 5: Sample DS/3 PRINT Dialog

```
-PRINT FICE, INSTNAME, CARNCODE, AAUPCODE, ENTOT, ENDRGTOT
   66 COLUMNS-(Y/F/B):
-PROOF
   F1            F2             F3     F4       F5         F6
  INST      INSTITUTION        CARN   AAUP   FALL 70    FALL 70
  FICE         NAME            CLASS FACSAL   TOTAL     DEG RES
  CODE                         CODE   CODE  ENROLLMENT GO TOTAL
 --6--- ----------25---------- --      --    ---,---    ---,---
 NO STAT FIELDS
 66 COLUMNS
 NEXT:
-SUM F5, F6
 66 COLUMNS
 NEXT:
-AVE F5, F6
 66 COLUMNS
 NEXT:
-PRINT SAME WHERE CARNCODE EQ 11 AND ENTOT GT 0

   66 COLUMNS-(Y/F/B):
-F
```

Chart 7: Typical DS/3 Output, Major Research Universities

```
INST            INSTITUTION           CARN   AAUP    FALL 70      FALL 70
FICE               NAME               CLASS FACSAL    TOTAL       DEG RES
CODE                                  CODE   CODE   ENROLLMENT   GD TOTAL
 1131 CAL INST OF TECHNOLOGY           11     0       1,512         742
 1305 STANFORD UNIVERSITY              11     7      12,566       5,386
 1312 U OF CAL BERKELEY                11     0      34,799       9,173
 1313 U OF CAL DAVIS                   11     0      16,556       2,296
 1315 U OF CAL LOS ANGELES             11     0      46,669       5,806
 1317 U OF CAL SAN DIEGO               11     0       9,386       1,189
 1328 U OF SOUTHERN CALIFORNIA         11     5      20,593       9,165
 1370 U OF COLORADO MAIN CAMPUS        11     5      21,482       3,496
 1426 YALE UNIVERSITY                  11     7       8,927       2,966
 1535 UNIVERSITY OF FLORIDA            11     4      23,903       4,061
 1739 NORTHWESTERN UNIVERSITY          11     7      15,571       4,159
 1774 UNIVERSITY OF CHICAGO            11     7       9,405       3,476
 1775 U OF ILL URBANA CAMPUS           11     5      36,109       8,458
 1809 INDIANA U AT BLOOMINGTON         11     5      30,403       7,675
 1825 PURDUE U MAIN CAMPUS             11     6      25,844       5,115
 1892 UNIVERSITY OF IOWA               11     5      20,604       4,802
 2029 TULANE U OF LOUISIANA            11     5       8,453       2,215
 2077 JOHNS HOPKINS UNIVERSITY         11     7       9,735       3,180
 2103 U OF MD MAIN CAMPUS              11     4      45,288       8,412
 2155 HARVARD UNIVERSITY               11     0      18,788       7,882
 2178 MASS INST OF TECHNOLOGY          11     7       7,787       3,679
 2290 MICHIGAN STATE UNIVERSITY        11     0      44,092       9,722
 2516 U OF MISSOURI COLUMBIA           11     0      22,592       4,307
 2520 WASHINGTON UNIVERSITY            11     5      11,062       2,294
 2627 PRINCETON UNIVERSITY             11     6       5,171       1,477
 2707 COLUMBIA U MAIN DIVISION         11     7      16,120       8,260
 2711 CORNELL U MAIN CAMPUS            11     6      16,163       3,810
 2785 NEW YORK UNIVERSITY              11     6      42,389      16,226
 2807 ROCKEFELLER UNIVERSITY           11     0         110         110
 2894 UNIVERSITY OF ROCHESTER          11     6       8,597       2,833
 2920 DUKE UNIVERSITY                  11     7       7,738       1,542
 2974 U OF NC AT CHAPEL HILL           11     5      17,567       3,922
 3024 CASE WESTERN RESERVE U           11     5       9,459       3,157
 3223 U OF OREGON MAIN CAMPUS          11     5      15,301       3,671
 3378 U OF PENNSYLVANIA                11     7      19,577       6,423
 3379 U OF PITTSBG MAIN CAMPUS         11     5      27,548       7,493
 3530 U OF TENNESSEE KNOXVILLE         11     4      27,299       4,714
 3658 U OF TEXAS AT AUSTIN             11     5      45,442       5,762
 3675 UNIVERSITY OF UTAH               11     5      21,687       3,438
 3798 UNIVERSITY OF WASHINGTON         11     5      33,585       6,983
 3895 U OF WISCONSIN MADISON           11     6      34,445       8,777
 3969 U OF MINN MNPLS SNT PAUL         11     0      60,679       8,309
 6883 OHIO STATE U MAIN CAMPUS         11     4      46,377       7,519
 6964 RUTGERS U NEW BRUNSWICK          11     0      26,192       5,956
 6965 PA STATE U MAIN CAMPUS           11     4      35,733       4,650
 9092 U MICHIGAN MAIN CAMPUS           11     0      36,967       9,475
        FALL 70      FALL 70
         TOTAL       DEG RES
      ENROLLMENT     GD TOTAL
CNT         46           46
SUM  1,056,262      244,163
AVE     22,962        5,307
 46 ENTRIES QUALIFY.
NEXT:
```

DS/3 is also capable of providing calculated values. Chart 8 shows a user's request for a calculation of graduate students as a percentage of total enrollment and tuition income per student. The dialog is shown on Chart 8, including commands for increasing the page width beyond the standard 72 characters and supplying more descriptive headings than a single line. Chart 9 shows the result of this inquiry.

DS/3 further provides the analyst with a capability of probing the data; that is, initial queries of the data may yield information that will stimulate further queries. Interacting with the data on an on-line basis provides the analyst with the luxury of deeper and deeper probes. An illustration and interpretation of a series of queries using DS/3 are presented in Chart 10.

These examples show the effectiveness of the DS/3 language operating on a properly structured data base. NCFPE analysts who used the DS/3 data base on a day-to-day basis appreciated this research capacity.

Chart 8: Typical DS/3 PRINT Dialog with Calculated Values

```
-PRINT SAME, X=1.00*(ENDRGTOT/ENTOT), Y=REGTUI/ENTOT
   93 COLUMNS-(Y/F/B):
-HEADING F7 PERCENT/GRADUATE
 93 COLUMNS
 NEXT:
-SUM F5, F6, F7, F8
 93 COLUMNS
 NEXT:
-AVE F5, F6, F7, F8
 93 COLUMNS
 NEXT:
-MARGIN 132 T
 NEXT:
-HEADING F8 TUITION/INCOME/'PER STUDENT'
 93 COLUMNS
 NEXT:
-PROOF
   F1          F2              F3    F4       F5        F6          F7              F8
  INST     INSTITUTION        CARN  AAUP   FALL 70   FALL 70     PERCENT          TUITION
  FICE        NAME            CLASS FACSAL  TOTAL    DEG RES     GRADUATE          INCOME
  CODE                        CODE  CODE  ENROLLMENT GD TOTAL                    PER STUDENT
 --6--- ----------25----------  --    --    ---,---   ---,---  +--,---,---.- +---,---,---
 F#   HEADING      CNT  SUM  AVE  MIN  MAX  PCT
 F5    FALL 70  /       ***  ***
 F6   FALL 70 /DEG      ***  ***
 F7   PERCENT /GRA      ***  ***
 F8     TUITION         ***  ***
 93 COLUMNS
 NEXT:
-PRINT SAME WHERE CARNCODE EQ 11 AND ENTOT GT 0
   93 COLUMNS-(Y/F/B):
-F


 INST         INSTITUTION          CARN  AAUP    FALL 70    FALL 70   PERCENT     TUITION
 FICE            NAME              CLASS FACSAL   TOTAL     DEG RES   GRADUATE     INCOME
 CODE                              CODE  CODE   ENROLLMENT GD TOTAL              PER STUDENT
  1131 CAL INST OF TECHNOLOGY       11     0       1,512       742       0.49         2,317
  1305 STANFORD UNIVERSITY          11     7      12,566     5,386       0.43         2,163
  1312 U OF CAL BERKELEY            11     0      34,799     9,173       0.26
  1313 U OF CAL DAVIS               11     0      16,556     2,296       0.14
  1315 U OF CAL LOS ANGELES         11     0      46,669     5,806       0.12
  1317 U OF CAL SAN DIEGO           11     0       9,386     1,189       0.13
  1328 U OF SOUTHERN CALIFORNIA     11     5      20,593     9,165       0.45         1,809
  1370 U OF COLORADO MAIN CAMPUS    11     5      21,482     3,496       0.16
  1426 YALE UNIVERSITY              11     7       8,927     2,966       0.34         2,361
  1535 UNIVERSITY OF FLORIDA        11     4      23,903     4,061       0.17           373
```

Chart 9: Probing a Data File with DS/3

```
-TALLY
 3110 ENTRIES QUALIFY    (Total number of students surveyed)
-TALLY WHERE SSFIRST EQ 2
 690 ENTRIES QUALIFY     (Number of students not attending their first
                            choice school)

-SUMMARY SSINCØME, SSTUIFEE, SSSTSCHL & SSØTHSCH

                SSINCØME       SSTUIFEE        SSSTSCHL & SSØTHSCH

ENT                 2973           3099               2983
SUM          $33,105,073     $3,936,338          2,420,132
AVE              $11,135         $1,270                811
MIN                   $0             $0                  0
MAX              $75,000         $4,000              4,400

      (Summary statistics for all students for family income,
        tuition and fees paid, and total scholarship dollar received)

-SUMMARY SAME WHERE SSFIRST EQ 2

                SSINCØME       SSTUIFEE        SSSTSCHL & SSØTHSCH

ENT                  671            689                661
SUM           $7,621,633       $838,988            462,557
AVE              $11,358         $1,217                699
MIN                   $0             $0                  0
MAX              $40,000         $3,470              4,200

      (Summary statistics for those students not attending their first
         choice school)

-TALLY WHERE SSFIRST EQ 2 AND SSWHYNF EQ 1
 292 ENTRIES QUALIFY    (Number of students not attending their choice
                           school because they were refused admission)
-TALLY WHERE SSFIRST EQ 2 AND SSWHYNF EQ 2
 273 ENTRIES QUALIFY    (Number of students not attending their first
                          choice school because it was too expensive)
```

```
SUMMARY SAME  WH SSFIRST EQ 2 AND SSWHYNF EQ 2 AND SSFCØNT EQ 1
AND SSACØNT EQ 2

               SSINCØME         SSTUIFEE       SSSTSCHL & SSØTHSCH

ENT                 108              109                 105
SUM          $1,368,580          $83,161              40,104
AVE             $12,672             $762                 381
MIN                  $0             $159                   0
MAX             $32,000           $1,900               2,570
```

(Summary statistics for those students not attending their first choice school because it was too expensive and whose first choice school was private and the school they are attending is public)

```
-SUMMARY SAME WH SSFIRST EQ 2 AND SSWHYNF EQ 2 AND SSFCØNT EQ 1
 AND SSACØNT EQ 1

               SSINCOME         SSTUIFEE       SSSTSCHL & SSØTHSCH

ENT                  61               63                  61
SUM            $726,835         $118,520              62,799
AVE             $11,915           $1,881               1,029
MIN              $3,923             $380                   0
MAX             $34,745           $3,000               3,750
```

(Summary statistics for those students not attending their first choice school because it was too expensive and whose first choice school was private and the school they are attending is private)

V. STATISTICAL ANALYSES

Since the DS/3 data management software operated on standard OS data sets, it was fairly easy to perform statistical analyses on the same data sets. SDC supported SPSS and BMD on their Time Sharing Option (TSO) system, and therefore the staff could access those packages at time-sharing terminals.

Chart 10 presents the input required by the user to set up a two-way cross-tabulation table for the ISSC student data file.* The input format for the data follows directly from the DS/3 file definition procedure, and the same mnemonic names for the variables in DS/3 were used to allow clearer interpretation of the output. Chart 11 illustrates the output for the cross-tabulation routine.

Chart 12 gives another example of SPSS input linking the STSCHLR student data file with the SPSS regression routine. Again, mnemonic variable names similar to the DS/3 names are used, and the input format corresponds directly with the DS/3 file definition for STSCHLR. The statistical results from the simple regression analysis are presented in Chart 13.

The data set as defined for use by DS/3 can be accessed through an on-line statistical package. No complicated reformatting or special file construction is needed. This feature gives the analyst the capability of performing more complicated statistical tasks with the data base, after initial queries and basic analyses have been performed, using a data retrieval language such as DS/3.

---

*This is a file from the Illinois State Scholarship Commission with data on each student who requested student aid.

Chart 10: Illustration of SPSS Input Commands (Cross-Tabulations)

```
-EDIT CØNTROL DATA NEW
 INPUT
 00010 RUN NAME          ANALYSIS ØF ISSC STUDENT DATA - NO. 2
 00020 VARIABLE LIST     ISPRØG, ISTYPCØL, ISYRNCØL, ISWATCØL, ISAGE, ISCUE
 00030 INPUT MEDIUM      DISK
 00040 # OF CASES        1294
 00050 INPUT FØRMAT      FIXED (4X, A1, 3X, 2A1, 5X, A1, 40X, A2, 1X, A1)
 00060 MISSING VALUES    ISPROG TO ISRACE (' ')
 00070 PRINT FØRMATS     ISPRØG TØ ISRACE (A)
 00080 CROSSTABS         ISTYPCOL BY ISWHTCOL
 00090 OPTIONS           1,5
 00100 STATISTICS        ALL
 00110 READ INPUT DATA
 00120 FINISH

-FREE F(FTOSF001)
-FREE F(FTOGF001)
-ALLØCATE F(FT05F001)    DA ('TSNUMBR. CØNTRØL. DATA')  SHR
-ALLØCATE F(FT06F001)    DA (*)
-ALLØCATE F(FT08F001)    DA ('TSNUMBR. DS3. ISSC')  SHR
-CALL 'SYS 2.SPSSLØAD (SPSS0272)'  '160000'
```

Chart 11: Illustration of SPSS Output

```
ANALYSIS OF ISSC STUDENT DATA                                            08/27/73         PAGE 2
FILE       NONAME      (CREATION DATE = 08/27/73)
* * * * * * * * * * * * * * * *C R O S S T A B U L A T I O N  O F* * * * * * * * * * * * * * * *
     ISTYPCOL                                BY ISWHTCOL
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *PAGE 1 OF 1
                 ISWHTCOL
-------------------------------------------------------------------------------------------------
          COUNT  I
        ROW PCT  I                                                           ROW
        COL PCT  I                                                           TOTAL
                 I
ISTYPCOL --------I   4    I   3    I   2    I   1    I   0    I--------I
    4            I   133  I     0  I     55 I   162  I      1 I   249  I  600
                 I  22.2  I   0.0  I    9.2 I  27.0  I    0.2 I  41.5  I 46.4
                 I  96.4  I   0.0  I   40.7 I  36.2  I  100.0 I  43.5  I
                -I--------I--------I--------I--------I--------I--------I
    3            I     1  I     1  I      2 I     0  I      0 I    12  I   16
                 I   6.3  I   6.3  I   12.5 I   0.0  I    0.0 I  75.0  I  1.2
                 I   0.7  I 100.0  I    1.5 I   0.0  I    0.0 I   2.1  I
                -I--------I--------I--------I--------I--------I--------I
    2            I     3  I     0  I     43 I     3  I      0 I    32  I   81
                 I   3.7  I   0.0  I   53.1 I   3.7  I    0.0 I  39.5  I  6.3
                 I   2.2  I   0.0  I   31.9 I   0.7  I    0.0 I   5.6  I
                -I--------I--------I--------I--------I--------I--------I
    1            I     1  I     0  I     34 I   282  I      0 I   264  I  581
                 I   0.2  I   0.0  I    5.9 I  48.5  I    0.0 I  45.4  I 44.9
                 I   0.7  I   0.0  I   25.2 I  63.1  I    0.0 I  46.2  I
                -I--------I--------I--------I--------I--------I--------I
                 I     0  I     0  I      1 I     0  I      0 I    15  I   16
                 I   0.0  I   0.0  I    6.3 I   0.0  I    0.0 I  93.8  I  1.2
                 I   0.0  I   0.0  I    0.7 I   0.0  I    0.0 I   2.6  I
                -I--------I--------I--------I--------I--------I--------I
         COLUMN      138        1       135     447         1     572     1294
          TOTAL     10.7      0.1      10.4    34.5       0.1    44.2    100.0

CHI SQUARE = 465.40161 WITH 20 DEGREES OF FREEDOM
CRAMER'S V = 0.29986
CONTINGENCY COEFFICIENT = 0.51432
KENDALL's TAU B = 0.17245
KENDALL'S TAU C = 0.13358
GAMMA = 0.26781
SOMER'S D = 0.16118
```

Variable definitions:

ISTYPCOL-type of college currently attending
1 = Public 4 year
2 = Public 2 year
3 = Private 2 year
4 = Private 4 year

ISWHTCOL-type of college the student would attend without scholarship

1 = Public 4 year
2 = Public 2 year
3 = Private 2 year
4 = Private 4 year

Chart 12: Illustration of SPSS Input Commands (Regression)

```
-EDIT CONTRØL2. DATA NEW
 INPUT
 00010 RUN NAME              ANALYSIS OF STATE SCHOLARSHIP DATA
 00020 VARIABLE LIST         SSINCØME, SSTUIFEE, SSSTSCHL
 00030 INPUT MEDIUM          DISK
 00040 # OF CASES            3110
 00050 INPUT FØRMAT          FIXED (11X, F5.0, 41X, F4.0, 40X, F4.0)
 00060 CØMPUTE               NEEDRAT = SSTUIFEE/SSINCOME
 00070 CØMPUTE               NEEDR2 = NEEDRAT * * 2
 00080 PRINT FØRMATS         NEEDRAT, NEEDR2, SSINCØME TO SSSTSCHL (0)
 00090 REGRESSIØN            VARIABLES = SSSTSCHL, NEEDRAT, NEEDR2
 00100                       REGRESSION = SSSTSCHL WITH NEEDRAT, NEEDR2 (2)
 00110 READ INPUT DATA
 00120 FINISH
```

Chart 13: Illustration of SPSS Output (Regression)

```
STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES, VERSION OF 02/01/72          01/24/74      PAGE 1
                   RUN NAME       ANALYSIS OF STATE SCHOLARSHIP DATA
                   VARIABLE LIST  SSINCOME, SSTUIFEE, SSSTSCHL
                   INPUT MEDIUM   DISK
                   # OF CASES     3110.
                   INPUT FORMAT   FIXED (11X,F5.0,41X,F4.0,40X,F4.0)
IGNORING INDEFINITE REPETITION, THE INPUT FORMAT PROVIDES FOR  3 VARIABLES.  3 WILL BE READ
IT PROVIDES FOR 1 RECORDS ('CARDS') PER CASE.  A MAXIMUM OF 105 'COLUMNS' ARE USED ON A RECORD
                  COMPUTE        NEEDRAT=SSTUIFEE/SSINCOME
                  COMPUTE        NEEDR2=NEEDRAT**2
                  PRINT FORMATS  NEEDRAT, NEEDR2, SSINCOME TO SSSTSCHL (0)
                  REGRESSION     VARIABLES=SSSTSCHL, NEEDRAT, NEEDR2
                                 REGRESSION=SSSTSCHL WITH NEEDRAT, NEEDR2(2)
                  READ INPUT DATA
ANALYSIS OF STATE SCHOLARSHIP DATA                                       01/24/74      PAGE 2

FILE   NONAME   (CREATION DATE = 01/24/74)
* * * * * * * * * * * * * * * * * *M U L T I P L E   R E G R E S S I O N* * * * * * * * * * * * * * * * * *
DEPENDENT VARIABLE..    SSSTSCHL
VARIABLE(S) ENTERED ON STEP NUMBER 1..  NEEDR2
                                        NEEDRAT

MULTIPLE R       0.41777       ANALYSIS OF VARIANCE    OF    SUM OF SQUARES      MEAN SQUARE       F
R SQUARE         0.17453       REGRESSION               2.  153602339.24356   76801169.62178  328.45929
STANDARD ERROR 483.55196       RESIDUAL              3107.  726486492.15066     233822.49506

-------- VARIABLES IN THE EQUATION -----------        ------ VARIABLES NOT IN THE EQUATION ----------

VARIABLE        B           BETA  STD ERROR B   F       VARIABLE     BETA IN     PARTIAL  TOLERANCE    F

NEEDR2      -245.30056  -0.38310 15.37293  254.615
NEEDRAT     1536.17696   0.60821 60.63940  641.760
(CONSTANT)   333.50195

ANN VARIABLES ARE IN THE EQUATION
ANALYSIS OF STATE SCHOLARSHIP DATA                                 01/24/74          PAGE 3
ALL
```

```
FILE       NONAME       (CREATION DATE = 01/24/74)
* * * * * * * * * * * * * *M U L T I P L E  R E G R E S S I O N* * * * * * * * * * * * * * * * * *

DEPENDENT VARIABLE. .        SSSTSCHL

                                   SUMMARY TABLE

VARIABLE                       MULTIPLE R   R SQUARE  RSQ CHANGE  SIMPLE R        B          BETA

NEEDR2                          0.06346     0.00403    0.00403    0.06346   -245.30056   -0.38310
NEEDRAT                         0.41777     0.17453    0.17050    0.32693   1536.17696    0.60821
(CONSTANT)                                                                   330.50195
ANALYSIS OF STATE SCHOLARSHIP DATA

                                                    01/24/74          PAGE 4


*****159664 BYTES FROM GIVEN SPACE NOT USED FOR REGRESSION******


            FINISH
ANALYSIS OF STATE SCHOLARSHIP DATA                 01/24/74          PAGE 5

     NORMAL END OF JOB.
     12 CONTROL CARDS WERE PROCESSED.
      0 ERRORS WERE DETECTED.

READY
```

VI. THE NATIONAL POSTSECONDARY EDUCATION FINANCING MODEL

In addition to analyzing the data in the extensive finance files gathered by NCFPE from numerous sources—using the capabilities of the DS/3 language and the Statistical Package for Social Sciences (SPSS)—the Commission staff developed a national financing model. This model provided enrollment impacts for various federal, state and local financing programs—by type of institution, by level of student aid, by family income level, and cost impacts by source of financing.

The model itself required one rather substantial, but unchanging data base and one small, but always changing file of data for the specific model run. To facilitate its use by the Commission, and hopefully later in Congress and state legislatures, the model was designed for interactive use.

One program was used to build model input for each run. This input program had an interactive dialog and was designed to minimize the input data requirements. Chart 14 shows the terminal user instructions provided by the input program and the first requests for data. Since a number of different people were using the model, a summary of the input instructions was included in the initial printout. Chart 15 shows some of the typical dialog. Since the input data requirements were different for each type of plan, the dialog was dependent on the type of plan.

At the conclusion, the input program requests the numbers of the output tables that should be printed at the terminal. Control is passed to the second program—the model itself. The model produces, from the two input files, an output file.

A third program, the report writer, uses the output file to produce the requested tables, in the sequence requested. Chart 16 shows a sample table—the summary table most often requested. Other tables provide more detail on enrollment and financing changes.

Chart 14: Terminal User Instructions from the National Postsecondary Education Financing Model

```
NCFP03              NATIONAL COMMISSION ON THE FINANCING           11/16/73
VERSION  1.2              OF POSTSECONDARY EDUCATION               14.26.55

          THIS MODEL ON THE FINANCING OF POSTSECONDARY EDUCATION
          WAS DEVELOPED BY THE NCFPE STAFF TO SUPPORT ITS
          ANALYSIS.  IN ADDITION TO THE INPUT PROVIDED BY THE
          USER, THERE ARE FILES OF EXTERNAL VARIABLES
          REPRESENTING DATA ON CURRENT AND PROJECTED ENROLLMENTS,
          FUNDING PATTERNS, AND STUDENT PRICE RESPONSE
          COEFFICIENTS.  ALSO THERE ARE ASSUMPTIONS INHERENT IN
          THE MODEL ITSELF.  USERS SHOULD BE AWARE OF THE UNDER-
          LYING ASSUMPTIONS BEFORE USING THE MODEL.

      BASIC INPUT INSTRUCTIONS:
          (1) VARIABLES SHOULD BE ENTERED AFTER THE "?" WITHOUT
             COMMAS AND WITH ONE OR MORE SPACES BETWEEN VALUES.
          (2) TERMINATING ANY LINE OF VALUES WITH A "/" PROVIDES
             ZERO VALUES FOR ANY REMAINING VARIABLES.
          (3) FOR THOSE VARIABLES REQUIRING A SEQUENCE OF LINE
             "0/" AS THE LAST VALUE.
          (4) NEGATIVE NUMBERS SHOULD HAVE A LEADING MINUS SIGN
             WITHOUT A SPACE.


ENTER A "1" IF THIS RUN IS A COMPLETELY NEW POLICY ALTERNATIVE
ENTER A "2" IF THIS RUN IS A CHANGE TO THE PREVIOUS RUN
?
1
ENTER NAME OF POLICY ALTERNATIVE        I
        (UP TO 40 CHARACTERS)           I
PLAN101-IB/SS WH PUBTU EQ 50% INS COST .
ENTER EXPANDED DESCRIPTION OF POLICY ANALYSIS                              I
        (12 LINES OF 72 CHARACTERS)                                        I
PUBLIC TUITIONS SET AT 50% OF INSTRUCTIONAL COSTS. NO ADDITIONAL...
```

Chart 15: Sample National Postsecondary Education Model Dialog

```
ENTER CUTOFF LEVEL OF INCOME FOR ACCESS ANALYSIS
?
10000/
ENTER TUITION LEVELS
INSTITUTIONAL CATEGORY (1-9), PERCENT OF TUITION INCREASE
GOING TO STUDENT AID, YEAR1 (19XX), TUITION IN $S, YEAR2,
TUITION IN $S, YEAR3, TUITION IN $S
?
1 0 1977 995 1980 1178/
?
2 0 1977 1016 1980 1204/
?
3 0 1977 1525 1980 1806/
?
4 0 1977 3049 1980 3611/
?
0/
ENTER FEDERAL INSTITUTIONAL AID
INSTITUTIONAL CATEGORY, CODE (0=LUMP, 1=CAPITATION,
2=PER STUDENT AIDED), YEAR 1, DOLLARS, YEAR 2, DOLLARS,
YEAR 3, DOLLARS
?
1 1 1977 0 1980 0/
?
0/
ENTER STATE INSTITUTIONAL AID
INSTITUTIONAL CATEGORY, CODE (0=LUMP, 1=CAPITATION,
2=PER STUDENT AIDED, YEAR 1, DOLLARS, YEAR 2, DOLLARS,
YEAR 3, DOLLARS
?
1 1 1977 400 1980 600/
?
2 1 1977 400 1980 600/
?
3 1 1977 400 1980 600/
?
4 1 1977 400 1980 600/
?
0/
```

It takes approximately twenty minutes to enter the data for an iteration of the model, less than fifteen seconds to compute, and about one minute per report for output. It is possible, then, to produce a simulation of a financing plan in about one hour after the details of the plan are completed. To assist users, a worksheet was developed organizing the data required by the model.

Thus, the NCFPE developed a data processing capability that permitted analysis of a large number of financing alternatives. The model became a vehicle for representing some of the complex interrelationships—such as student price response and enrollment induced institutional costs—that could not be accommodated in simple analysis. This national postsecondary education financing model will be described more fully along with the analytical framework developed by the Commission in a separate staff report.

Chart 16: Sample Output of the National Postsecondary Education Financing Model

```
NCFP05           NATIONAL COMMISSION ON THE FINANCING           PAGE 2
VER 1.2             OF POSTSECONDARY EDUCATION                  11/16/73
RST                                                             14.26.55

     ANALYSIS OF PLAN101-IB/SS WH PUBTU EQ 50% INS COST

                        TABLE 1 - SUMMARY
                                               1977       1980       0
I.   PROPOSED POLICY CHANGES

ADDITIONAL STUDENT AID, FEDERAL                  0.         0.       0.
   MAXIMUM ELIGIBLE FAMILY INCOME            10000.     10000.       0.
ADDITIONAL STUDENT AID, STATE                    0.         0.       0.
   MAXIMUM ELIGIBLE FAMILY INCOME            10000.     10000.       0.
ADDITIONAL INSTITUTIONAL AID, FEDERAL            0.         0.       0.
   (NONE SPECIFIED)
ADDITIONAL INSTITUTIONAL AID, STATE            400.       600.       0.
   (CAPITATION)
FEDERAL TRANSFERS, LOCAL GOVERNMENT              0.         0.       0.
FEDERAL TRANSFERS, STATE GOVERNMENT              0.         0.       0.
*  TOTAL FEDERAL TRANSFERS                       0.         0        0.
AVERAGE TUITION BEFORE STUDENT AID
     PUBLIC  2 YEAR                            995.      1178.       0.
     PUBLIC  4 YEAR, LOWER DIVISION           1016.      1204.       0.
     PUBLIC  4 YEAR, UPPER DIVISION           1525.      1806.       0.
     PUBLIC  4 YEAR, GRADUATE                 3049.      3611.       0.
     PRIVATE, UNDERGRADUATE                   2039.      2415.       0.
     PRIVATE, GRADUATE                        2039.      2415.       0.
NONCOLLEGIATE                                 1326.      1570.       0.

II.  PROJECTED PERCENT CHANGES FROM FORECAST ENROLLMENT

PUBLIC  2 YEAR                              -11.55 %    -13.67 %   0.0 %
PUBLIC  4 YEAR, LOWER DIVISION               -4.86 %     -5.76 %   0.0 %
PUBLIC  4 YEAR, UPPER DIVISION              -13.91 %    -16.47 %   0.0 %
PUBLIC  4 YEAR, GRADUATE                    -32.67 %    -38.69 %   0.0 %
PRIVATE, UNDERGRADUATE                        1.82 %      2.15 %   0.0 %
PRIVATE, GRADUATE                             3.62 %      4.28 %   0.0 %
NONCOLLEGIATE                                 2.63 %      3.12 %   0.0 %
UNDERGRADUATE, LESS THAN $10,000             -8.32 %     -9.85 %   0.0 %
UNDERGRADUATE, $10,000 TO $14,999            -4.60 %     -5.46 %   0.0 %
UNDERGRADUATE, $15,000 AND OVER              -2.70 %     -3.21 %   0.0 %

III. PROJECTED FINANCING CHANGES

TOTAL INCREMENTAL COST, FEDERAL          -370913792.  -457498624.       0.
TOTAL INCREMENTAL COST, STATE           -4359847940. -4687699970.       0.
TOTAL INCREMENTAL COST, LOCAL              -4830495.    -6052100.       0.
*  TOTAL GOVERNMENTAL INCREMENTAL COST -4735590400. -5151248380.       0.
TOTAL INCREMENTAL COST, PRIVATE             2686256.    27366368.       0.
TOTAL INCREMENTAL COST, FAMILY & STU     -283298304.  -397837056.       0.
** TOTAL INCREMENTAL COST               -5016199170. -5521715200.       0.
ADD TUITION REVENUE FOR STUDENT AID     -2297455360. -2793006340.       0.
```

VII. USERS OF THE DATA BASE

There are many potential users of the NCFPE data base. For example, representatives of the Association of Independent California Colleges and Universities (AICCU) in Los Angeles and the National Center for Higher Education Management Systems (NCHEMS) in Boulder, Colorado, requested permission to use the NCFPE data base at the SDC Santa Monica facility. Arrangements were made through SDC for these two organizations to use the data base.* AICCU has created a special data base for California and made arrangements to use some of the student survey responses (STSCHLR and SRS). Their staff members believe that this use has significantly improved their research capabilities. Similarly, NCHEMS is using the data base for developing their national planning model.

In addition, the California Department of Finance and the Nebraska Legislative Fiscal Office requested demonstrations and used the system briefly to answer some of their questions.

Already, such uses of the NCFPE data base indicate the potential for improving the capability of a number of postsecondary education organizations and agencies in gaining access to useful information. The value of information like HEGIS, when it becomes immediately and easily available, has also been demonstrated. At the same time, NCFPE used the technology to protect data when sources placed restrictions on them.

Perhaps this experience will be the forerunner of a national data base immediately accessible to all potential users through terminals.

---

*At no cost to the Commission, both groups contracted for DS/3 and TSO service and were given file descriptions.

VIII. THE COSTS OF MAINTAINING AND USING THE DATA BASE

It is difficult to determine the costs of a system like the one described here. The costs are more a function of the specific charging algorithm* than any system characteristics. Chart 17 describes some of the data services and costs of the two facilities that the Commission used, HEW's Data Management Center and SDC's Corporate Computer Facility.**

The costs per terminal hour at the SDC facility are dependent upon the skill of the user, the type of retrieval, and type of terminal used. The costs varied from $14 to $157 per terminal hour initially. By using much larger blocking factors, different LOGON procedures, and separating DS/3 usage from TSO/Fortran, the cost per terminal hour decreased sharply. In addition, a new version of DS/3 released in September 1973 improved response time and cost less. It appears that the cost for DS/3, at the SDC facility, would be $20 to $35 per terminal hour. The price at DMC is fixed at $25 per terminal hour. But since their computer was not a virtual machine and there were few users, the cost must be considerably more that the price. Further experience by users should reduce costs significantly at either facility. It is not clear just what the lower limit of costs could be with a large number of users accessing a large shared file.

---

*Formula relating the uses of each computer resource with a price.

**Subsequent to this paper, SDC announced a 102.6% price increase for storage, which significantly affects the costs given here and which likely would have caused the Commission to have selected another vendor.

Chart 17: Comparative Data Services and Costs

		DHEW/DMC		SDC
Monthly File Storage (Megabytes)	22.8	$345.00	80.8	$808.00
Per Megabyte		$15.13		$10.00*
File Security		Password		Double Password
DS/3 Service				
Per Terminal Hour	150 bps	$25.00	300 bps	$20-35.00*
Per Table		$3.10		$1.80
TSO Service				
Per Terminal Hour		Unavailable		$15-50.00*
NCFPE Model Per Iteration		Unavailable		$10.00*
Statistical Analysis Per Terminal Hour		Unavailable		$50.00*

---

*Estimated average, largely dependent upon skill of the user. SDC increased the storage rate 102.6% to $20.26 per megabyte effective February 15, 1974.

IX. THE FUTURE OF THE DATA BASE

The experience of the National Commission has demonstrated that a national data base is economically feasible. In fact, the data base would save considerable expenditures on the part of researchers and analysts who otherwise would have to locate tapes and write special programs in order to have access to the data.

But more important, access is provided to data in minutes rather than months. While this timesaving is, in part, the result of telecommunications and data processing technology, it is achieved by determining the policy for data access in advance rather than on an individual, case by case basis. In addition, data processing becomes the responsibility of the user of the data rather than the provider of the data.

If policy analysis is to affect the course of decision making, it will have to be responsive to the policy makers decision-making time frame. While it is not possible to determine how responsive policy analysis must be, some congressional staff members have indicated that they usually need access to information within forty-eight hours. After that, the legislative decisions have usually been made. If this time frame is the basic criteria for policy analysis, then clearly this kind of technology and data base will be required for future decision making.

These considerations would suggest the need for a federal data base. But some practical problems in implementing such a national data base remain. First, the data access is accompanied by data processing services. It is not clear that the federal government could, or should, be a major provider of data processing services in competition with private industry. Second, some potential users have expressed concern about the objectivity of a federal data supplier. Third, emphasis will have to be placed on making data available, not on protecting data from potential misuse. There is no known way that a data provider can preclude misuse of data without severely limiting access.

In short, the Commission's experience has demonstrated the technical and economic feasibility of a nationally maintained data base. But there are a number of practical questions and policy issues that must be resolved before full implementation of a federally-maintained data base can be achieved. It is not clear at this time who will take that leadership.*

---

*In December 1973, Dr. John Ottina, U. S. Commissioner of Education, indicated that the National Center of Educational Statistics will take over the data base maintenance and permit public access like that provided by NCFPE to AICCU, NCHEMS, and the states of California and Nebraska.

DATA ENTRY TECHNIQUES

Haymond H. Satterfield
Data Entry Representative
IBM Corporation
New Orleans, Louisiana

DATA ENTRY TECHNIQUES

This session covered the historical base and current trends of data entry systems.

Historical facts for the United States indicate that the productivity of data entry methods increased approximately 25% from 1960 to 1970, however, wages for this productivity increased approximately 50% and volume increased approximately 500%. Therefore, data entry total costs have increased tremendously which should indicate that the need to study new methods of data entry is very important.

The key point of most of IBM's recent machines or systems announcements in the data entry field is to approach data entry on a distributive basis, that is, not all key data entry need be centralized independent of the source. A discussion concerning the importance of source data knowledge by the operator followed. The issues of economics, accuracy, speed, peak loads, etc. were discussed in centralized vs distributive data entry.

An overview of IBM's data entry devices was presented at the end of this session.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

NATIONAL SURVEY ON COMPUTER BASED MIS IN HIGHER EDUCATION

Richard L. Mann
Director
Management Information Systems
University of Illinois

A Summary of the Results of the Study of Computer Based Management Information Systems (CBMIS) in Higher Education

The purpose of this study was to determine the extent to which computer based management information systems (CBMIS) are being developed and used by colleges and universities throughout the country. The following definition of a computer based management information system (CBMIS) was used in this study: A computer based management information system (CBMIS) is used to store, manipulate, and retrieve data for management planning and resource allocation purposes. The CBMIS is distinguished from standard data processing applications such as payroll or student records in that it emphasizes the capability to rapidly integrate and display data from various files, both current and historical, to assist administrators in planning, resource allocation and general management decisions.

This study was based upon the completion of a survey questionnaire mailed to 722 institutions throughout the continental United States. Responses were received from over 66% of the institutions.

The study explored seven major areas:

I. To what extent are academic institutions developing CBMIS?

II. What factors have caused academic institutions to begin development of CBMIS?

III. How is the CBMIS project organized?

IV. What role does CBMIS play in institutional management and reporting?

V. What are the components of CBMIS in higher educational institutions?

VI. What impact has the Western Interstate Commission on Higher Education, National Center for Higher Education Systems (WICHE-NCHEMS) had upon CBMIS development in academic institutions?

VII. What is the impact of CBMIS on university administration?

A number of institutional demographic characteristics were used in analyzing the data, these included:

1) Campus type with categories a) single campus, b) system office for a multi-campus system, and c) campus in a multi-campus system; 2) Enrollment with categories a) 3000-6000, b) 6001-10,000. c) 10,001-15,000, d) 15,001-20,000, and e) 20,001 and over; 3) Control with categories a) public and b) private; and 4) Highest degree offered a) Associate, b) Bachelors-Masters, c) Doctorate.

I. Extent to Which Academic Institutions are Developing CBMIS

Of 430 institutions responding to the question "Is your institution planning or implementing a CBMIS?" 69% replied affirmatively. Forty per cent of these institutions reported their CBMIS in partial operation. 28% indicated they were still in the planning stage and less than one per cent of the schools stated that they had a fully operational CBMIS. More than half of the institutions reporting no CBMIS activity indicated that they intended to develop a CBMIS in the future.

Although no statistically significant differences relating to CBMIS development were observed when the data was analyzed by demographic characteristics, a number of interesting tendencies were observed: multi-campus institutions and public institutions reported CBMIS activity more frequently than did other institutions; schools with larger enrollments, especially over 20,000, showed a slightly greater percentage of CBMIS involvement; and it was observed that the higher the level of degree offered the greater the percentage of institutions which were engaged in CBMIS development.

When the length of time CBMIS institutions have been planning and operating their systems was analyzed, the data showed that public institutions have been planning and implementing CBMIS somewhat longer than private schools; multi-campus institutions have been engaged in CBMIS planning significantly longer than single campus schools, although single campus schools reported significantly more progress in terms of actually implementing CBMIS; the larger the enrollment the longer the institution was likely to have been engaged in both planning and implementing CBMIS; and in both public and private institutions the greatest percentage of schools reported that they began their CBMIS planning and implementation activities between 1 and 3 years ago.

The survey responses suggest that the longer an institution has been using a computer for administrative purposes the more likely it is that the school is planning or operating a CBMIS. However, little relationship between the length of time an institution has been engaged in administrative computing and the length of time it has been planning or operating a CBMIS could be found. Finally, the development of CBMIS appears to be independent of whether the host computer is dedicated to administrative work or is a shared administrative/instructional machine.

II. Factors Resulting in CBMIS Development

Respondents were asked to rank the most important reasons for their institutions' beginning CBMIS development. "To improve internal management" stood out as the most important with 229 institutions out of 293 ranking this reason first. The next most important reason specified was "support of other management tools" followed by "to meet state reporting requirements", which was ranked high by public institutions.

The Vice President for administration, planning or finance appeared to be the individual cited most frequently as the primary initiator of the CBMIS project. He was followed in importance by the President, Director of Data Processing and the Director of Institutional Research. Private institutions, however, reported the Data Processing Director as most responsible for initiating the CBMIS project followed by the Vice President, the Director of Institutional Research and then the President. The President, however, was cited most frequently as the highest level of administrative support which the CBMIS project had received.

III. Organization of the CBMIS Project

Each institution was asked to identify by title the administrator with overall responsibility for the CBMIS project. The most frequently mentioned administrator was the Director of Data Processing followed by the Director of Institutional Research/Planning, and the Vice President or Director of Finance or Business Affairs. Forty per cent of the responding institutions indicated that the chief administrator for the CBMIS reports directly to the President while another 34% indicated that he reports to the chief financial, business or planning officer of the school.

Responsibility for the design of CBMIS was assigned to the Data Processing department according to 53% of the respondents. Twenty-three per cent of the respondents indicated that CBMIS design was the responsibility of the office of Institutional Research/Planning, while 12% of the schools reported that this was the responsibility of a specially created task force or committee. In terms of implementation responsibility, Data Processing was mentioned most frequently (63%) followed by Institutional Research/Planning (19%) and special committees (10%). It appears that data processing has a predominant role in both CBMIS design and implementation.

The survey also showed that institutions developing CBMIS regard administrative support offices such as the Registrar, Personnel and Business Affairs as most important in terms of their contribution to the planning of a CBMIS followed closely by general level administrators such as the President and Vice Presidents. Academic administrators were ranked a poor third in importance and faculty and student contributions were reported as being negligible.

Slightly more than half of the responding institutions indicated that no additional funds were being allocated for CBMIS. Multi-campus system offices, however, reported a significantly higher frequency of new funding for CBMIS than was the case for other respondents. It was also noted that institutions with large enrollments (over 15,000) were more likely to create new budgets for CBMIS than were smaller schools suggesting that the smaller the institution the less likely it is to provide additional funds for CBMIS.

IV. CBMIS in Institutional Management and Reporting

Slightly less than half of the responding institutions indicated that they engaged in a formal study of information requirements prior to beginning CBMIS development. However, smaller institutions (under 6,000) showed a greater tendency to engage in such formal studies as compared to larger schools.

The survey sought to determine the extent to which major information areas (student, financial, staff and physical facilities) are being incorporated into the CBMIS. Student information appeared to be the most advanced followed by financial, staff and physical facilities information. A number of respondents also indicated that curriculum, alumni and library information areas were also being included in their CBMIS.

Private institutions appear to have progressed somewhat further in incorporating financial information into their CBMIS as compared to their public counterparts. A similar situation existed for student information, but physical facilities information has received greater attention from public institutions. No difference between public and private schools could be discerned in terms of their inclusion of staff information into their CBMIS.

The extent to which the computer is being used to maintain and process data for the four information areas was also explored. The order of development was the same as in CBMIS development: Student, Financial, Staff and Physical Facilities Information. As would be expected, the greater the extent of computer use, the more advanced the CBMIS development reported for each area. The media used in the computerization of these information areas appeared to be primarily direct access devices followed by magnetic tape and punched cards.

Respondents were queried as to the order in which various groups within the institution were serviced by CBMIS. As might be expected it was found that the design priorities built into the CBMIS reflected the extent of each group's contribution to the planning of the CBMIS. Thus, administrative support offices followed by general administrative offices, academic administration, faculty and students comprised the priority ranking for CBMIS service. This was the same as the ranked importance of each group's contribution to the planning of the CBMIS and which suggests that a CBMIS is primarily designed by administrators for use by administrators.

The second most important reason specified by respondents for developing CBMIS was the support of other management tools. The use of management tools such as resource allocation models was reported by 127 of the 296 institutions indicating CBMIS activity. Planning, Programming and Budgeting Systems (PPBS) were reported in use by 110 of the CBMIS schools while 205 of these institutions reported using institutional research techniques. Thus it would appear that the support of other management tools is in fact a major reason for many institutions' embarking upon CBMIS development.

V. Components of CBMIS

Four major components appear to constitute most CBMIS in academic institutions. These include integrated data bases, commonly defined data elements, generalized information retrieval systems, and techniques used to insure data security.

Most respondents reported that their CBMIS data base was composed of related sub-files which conceptually form a single integrated file. Few institutions indicated that they use either a single physically integrated file or separate unintegrated files for their CBMIS data base. In addition, most institutions reported that they are using operational data processing files (such as payroll and student records) as well as specially created summary files to support the CBMIS data base. A somewhat smaller group of schools reported using only operational data files while very few institutions indicated they are using only summary files.

Most CBMIS institutions reported that many of their data elements were commonly defined, followed by a slightly smaller percentage of schools which indicated that all data elements in the CBMIS were commonly defined, and several institutions in which only a few data elements were commonly defined. Interestingly, institutions in the 3,000-6,000 enrollment range reported the greatest frequency of all data elements commonly defined which may be due to either the ease of achieving such commonality in a small institution or the relative newness of the institution to administrative computing, enabling it to begin with commonly defined data elements.

The standards used to define common data elements differed significantly between public and private schools. Public institutions reported that WICHE-NCHEMS definitions followed by state or system-wide definitions were used while private schools indicated they used their own definitions most often followed by those of WICHE-NCHEMS. Interestingly, the same percentage (36%) of public and private institutions report using WICHE-NCHEMS definitions.

One method of controlling data element commonality in a CBMIS is through a data element dictionary. Most institutions reported that this feature of their CBMIS was still in the planning stage while a slightly smaller percentage reported that they were implementing or had implemented such a dictionary. Although very few respondents indicated they were not considering such a system, this response was more prevalent among single campus schools as opposed to multi-campus institutions. This fact suggests the importance which multi-campus schools attach to controlling data element definitions in order to manage multiple campuses.

A majority of the responding CBMIS institutions reported planning a generalized information retrieval system for their CBMIS. A somewhat smaller group indicated that such a system was already in operation while very few respondents indicated that they were not considering such a system.

Large institutions, over 15,000 students, showed a significantly greater frequency of systems already in operation. A majority of the institutions reported that their information retrieval systems run both on-line and batch requests and can handle scheduled as well as unanticipated reports. Most information retrieval systems also require the assistance of a professional programmer and do not have the capability to retrieve historical or diverse information as easily as current operating information from within a single information area.

Almost all of the responding CBMIS institutions indicated that administrative policies and data processing procedures exist to insure data security witnin the CBMIS data base. A majority of schools also indicated that software security systems were in use while only a few institutions reported that hardware security features were part of their CBMIS.

Of the 296 institutions reporting CBMIS activity, nearly half indicated that all four components (integrated data base, commonly defined data elements, generalized information retrieval system, and data security techniques) were included in their CBMIS. An integrated data base and commonly defined data elements were mentioned most frequently by respondents as being included in their CBMIS.

The use of commerical software products to support CBMIS development appeared to be limited. IBM's Information Management System (IMS) was the most frequently mentioned data management system while Informatic's MARK IV File Management System was the most often mentioned retrieval system.

VI. Impact of WICHE-NCHEMS

The impact of WICHE-NCHEMS on CBMIS development was found to be considerable. Although only a small percentage of the respondent institutions participated or NCHEMS task forces or attended implementation conferences, attendance at NCHEMS seminars and subscription to

NCHEMS publications was reported by a large percentage of the respondents. Public institutions with large enrollments tended to be the most influenced by WICHE-NCHEMS products. The Resource Requirements Prediction Model (RRPM) and NCHEMS data element definition standards appeared to be used heavily in CBMIS data bases. Finally, when asked to assess the impact of WICHE-NCHEMS upon the development of their CBMIS, respondents reported an overall positive influence, although private institutions appeared somewhat less positively impacted by NCHEMS activities than was the case for public schools.

VII. CBMIS Impact on University Administration

The use of CBMIS in nearly half the institutions reporting tends to be accompanied by some increase in the centralization of administrative decision making. A large majority of respondents also indicated that decisions which reflect objective management considerations have increased in comparison to those which reflect political considerations since the use of CBMIS began.

Organizationally, it appears that the introduction of CBMIS has had little effect upon the number of levels in the administrative hierarchy, has resulted in increases in the size of data processing and institutional research/planning staffs, has not affected administrative support staffing (Business Office, Registrar, etc.) at most institutions, and has resulted in overall administrative staff increases and increases in the number of administrators with quantitative or computer background.

Each respondent was asked to give his assessment of the overall impact which CBMIS is having or will have upon his institution. Most frequently mentioned was the ability to make better management decisions through the provision of accurate and readily accessible information. Emphasis was also placed upon expected improvements in institutional planning, resource allocation and multi-campus management. Although a number of institutions described difficulties which they were having in developing their CBMIS, most respondents indicated positive expectations. Perhaps the most supportive of all the comments was made by one respondent who described the impact which CBMIS has had upon his institution this way: "It's survival!"

ROUNDING SECOND BASE WITH DATA BASE

Glenn B. Harvey
Director of Computer Services
DePaul University

ROUNDING SECOND BASE WITH DATA BASE
DePAUL UNIVERSITY'S EXPERIENCE

Glenn B. Harvey
Director of Computer Services
DePaul University

This paper logically follows an earlier treatment of this subject entitled GETTING TO FIRST BASE WITH DATA BASE. While the over-all system in both papers is the same, the former placed emphasis on the batch maintenance aspects and this one targets on the methods for retrieval of information.

INTRODUCTION - DePaul's Early Computer Environment

DePaul University has always followed a tight budget philosophy toward computers and data processing. This is altogether consistent with the University's goal of providing a cost effective program in higher education.

DePaul has approximately 10,000 students, over 525 faculty and over 40,000 graduate alumni. An additional 250,000 people have taken educational programs of some duration at the University.

The 1972-73 operating budget is approximately $16.5 million, nearly triple the $5.5 million budget of 1963-64.

Strict control on the funding of computers paid financial dividends but left a widening instructional gap for DePaul graduates who learned little about the electronic marvel. On the administrative side, second generation IBM 1401 equipment, software and concepts were relied upon, even into the 1970's when surrounding institutions were progressing from third to fourth generation philosophies. Because operations staff and programing resources were also held to a minimum, this effectively curtailed DePaul's ability to respond to changing needs.

Pressures from both internal and external sources became stronger to expand the computer resources. By late 1971, this resulted in the acquisition of new high-level staff experienced in the application of computers to both administrative and instructional systems and dedicated to the proposition that computer systems should service users in unique new ways.

DECISION TO GO DATA BASE

To lend continuity and control to Systems Planning and Development efforts, a Master Plan for University-wide Computer Services was written and presented to the Executive Committee for approval. This plan outlined the alternatives for computer services available to DePaul, their associated costs and benefits.

Included were recommendations for hardware, software, staff and an implementation schedule for application systems. Objectives to be achieved by new systems required that they be:

- * Responsive - To changes in user needs regardless of timing or source of change.
- * Flexible - Able to provide for new files, new data in old files, new methods of processing old data, entire new systems, or whatever need arises.
- * Simple - Easily understood by users of data, as well as the computer technician.
- * Self-documenting - System should provide relatively complete documentation of itself as a by-product of its own operation.
- * Parameter-driven - Most functions to be handled by entry of control cards or tables with tailored programming held to a bare minimum.
- * User-controlled - User should have high degree of control over his own data files, their content, maintenance and usage.
- * Easily-converted - Little difficulty converting existing 1401 files from old to new systems.
- * Economical - All of the above goals should be possible on relatively low-cost computer equipment with small memory and few input/output units. Staff expenditures should likewise be minimal yet able to accommodate system demands and changes.

Hardware selected was an interim System 360/30, 96K memory, 4 tape/3 disk, computer to be followed by an IBM System 370/135, with the same memory and input/output units, when available in January 1973. The 3 disks were originally 2319's replaced by double-density ITEL disks in August, 1972, thus making approximately 175 million bytes of storage available on-line. The interim hardware was installed March, 1972.

It was recommended and approved that the new machine system be constructed using Data Base concepts. That is, files within the "data base" would be logically inter-related in a network fashion. This had the obvious advantage of permitting the development of application systems that could easily retrieve new "relationships" of data, i.e., new information for new uses, rather than just the data themselves. Traditional file-oriented machine systems are programmed to contain specific data relationships that satisfy needs at a point in time, but are time consuming and costly to reprogram when changes occur in the needs for information.

SELECTION OF DATA BASE TOOLS

Several decisions were also made in regard to existing computer software that could aid DePaul's development of a Data Base system. Much time was devoted to seeking out and evaluating "packages" that purported to cure an organization's data processing problems in information retrieval, report writing, total information systems, management information systems, data base management, data management or file management systems. Some significant discoveries were made.

This paper does not recommend nor condemn any given "package." We looked into many software aids that were rejected, for DePaul's use, for one reason or another. Included in this study were products and names such as ASSIST, BUG, Burrough's DISK FORTE, DATA ANALYZER, IBM's IMS & EPIC, MARK IV, Meta-COBOL, PRISM, RSVP, SCORE III, Stanford's "OASIS," SERIES, WORK TEN and others. Many of these products have excellent features, but still did not meet our criteria. This sometimes meant they cost too much. While a separate paper could be written on such an evaluation, several observations should be made here:

* Price is often negotiable, or subject to change.

* No one really knows what software is worth.

* A single user's recommendation for a package is never enough...sometimes several isn't enough.

* Good, cheap packages exist that are virtually undiscovered and unknown.

While investigating one such product (a data base management system) called TOTAL, we learned of its use at Amherst College in Amherst, Mass. Amherst had developed a sophisticated generalized maintenance system using TOTAL as the "super-access" method for organizing its disk files. Their approach to data processing problems appeared to us more advanced, responsive, flexible, etc. than any other we found. That system (or systems), used with the TOTAL package, seemed to be DePaul's answer to new system development. The Amherst system became the foundation for the DePaul system herein described. Only because Amherst provided their programs, documentation, staff time and patience while we were learning their operation, has the DePaul capability been made possible.

We believe that this has been one of the best examples of sharing computer systems between institutions that one can find. All of DePaul's enhancements to the system have, in return, been given to Amherst in token payment. Cost of the base system actually included (1) acquisition of the TOTAL package from its vendor, Cincom Systems, Inc. of Cincinnati, (2) staff time and expenses to visit Amherst on several occasions, (3) staff and machine time to study, analyze and modify the system from RCA to IBM computers and slightly different versions of COBOL, and (4) staff and machine time to enhance, in some cases rewrite, original programs to do more using different techniques.

Other software products DePaul acquired at this time included a COBOL Program Generator called SNAP (Simplified Notation for Application Programmers) that permits us to produce accurate, low-core, cross-reference COBOL D programs in a quick and easy manner. All of the possible features and benefits of this excellent, low-cost (under $3000) package have yet to be used by DePaul.

For our entry into the world of teleprocessing we selected a package called CRT-Interface, marketed by Westinghouse Tele-Computer Systems of Pittsburgh. This package requires minimal memory for operation (32K partition), permits COBOL programming of TP jobs and cost DePaul $900. To facilitate our data base back-up, we also purchased the Westinghouse Dump/Restore disk-to-tape program, for $800, the best program of its kind.

DATA BASE SYSTEM STANDARDS

The DePaul System is really a Data Management System. It is composed of four basic parts:

- * Generalized Maintenance System (GMS)
- * Generalized Retrieval (and Reporting) System (GRS)
- * Generalized Utility System (GUTS)

The fourth part is being added right now and permits TP inquiry/update into the Data Base -- to be called Generalized Terminal System (GTS). This paper treats only the second system, GRS, our method of retrieving information from the data base files.

GMS Job Stream

A single job stream GMJ001 exists for the updating, ala maintenance, of any and all files in the Data Base. All files can be active in the same run, that is, incoming transactions can be targeted for any Data Base file. It is expected that only one GMS run will be required each day to maintain all such files. The System Flow in Figure 1 shows input transactions being converted to disk via the IBM card-to-disk utility. Program GM0020 edits each transaction thoroughly, flags those with errors and releases them all to the update program GM0030. GM0030 is a calling program that calls the generalized module XX0900 to process all direct changes to the appropriate Data Base files.

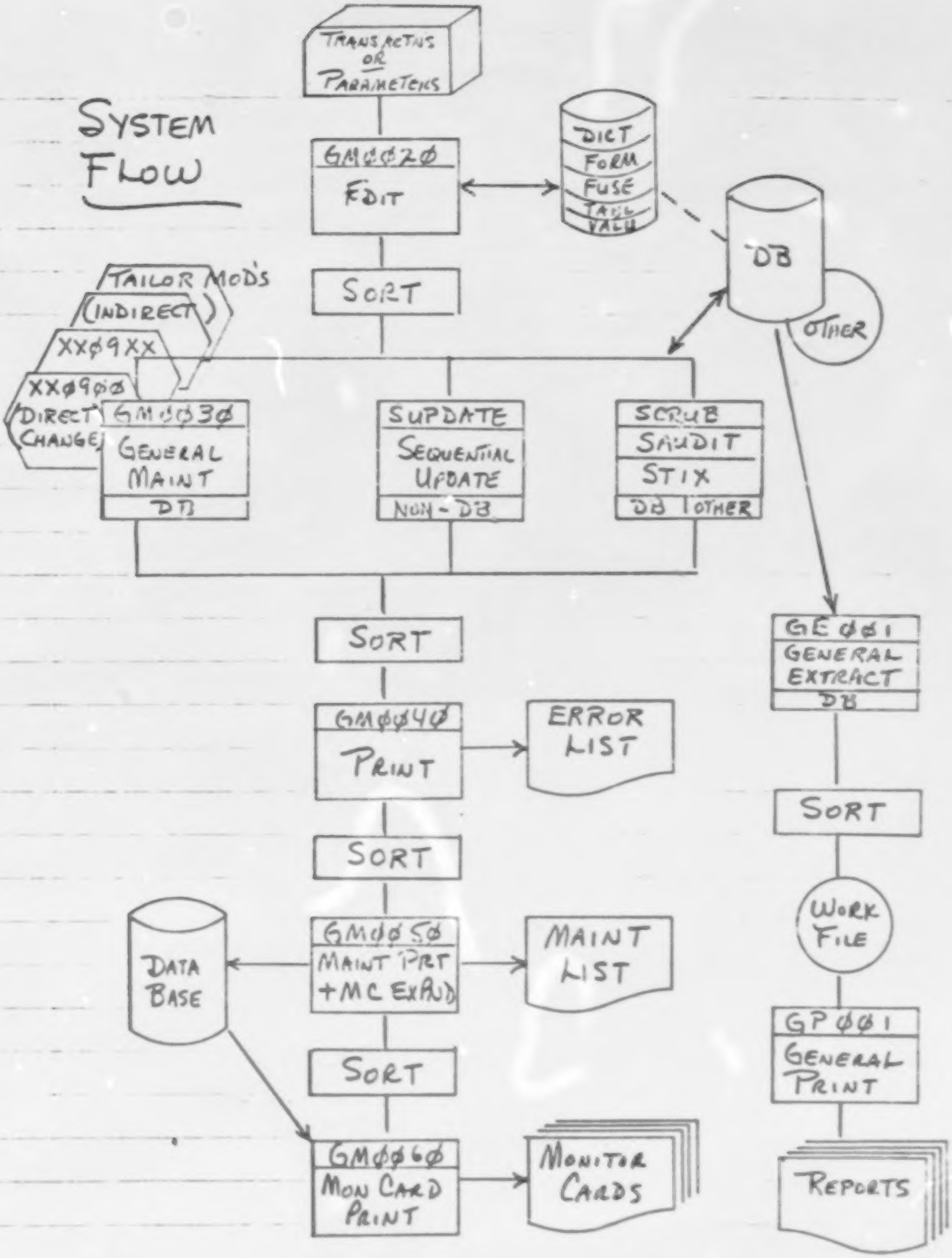

SYSTEM FLOW
TRANSACTNS OR PARAMETERS
GM0020
EDIT
DICT
FORM
FUSE
TABL
VALU
DB
OTHER
SORT
TAILOR MOD'S (INDIRECT)
XX09XX
XX0900
DIRECT CHANGES
GM0030
GENERAL MAINT
DB
SUPDATE
SEQUENTIAL UPDATE
NON-DB
SCRUB
SAUDIT
STIX
DB | OTHER
SORT
GM0040
PRINT
ERROR LIST
SORT
DATA BASE
GM0050
MAINT PRT +MC EXPND
MAINT LIST
SORT
GM0060
MON CARD PRINT
MONITOR CARDS
GE001
GENERAL EXTRACT
DB
SORT
WORK FILE
GP001
GENERAL PRINT
REPORTS

FIGURE 1

STANDARD DATA BASE TERMINOLOGY

- System No. = AA 2-digit alpha
  - System Logical File = Usually 1 mast file with some linked Var Files
  - Such as RG = Registrar
    AD = Admissions
- File No. = AA (Standard Abbreviation) or
  AAAA ('Total' Name)
  - Physical File = Master or Variable
  - Such as PE = PEOP
    CR = COUR
- Data Element No. = AA (File Abbreviation)
  NNN (Element #)
  - Identified by Physical File
  - AA001 required for Master or Variable Files and = Control # (up to 30)
  - Such as PE001 = PEOP SS#
    PE010 = PEOP Last Name
  - Can be redefined in DICT to designate combinations of other elements or the unique processing of the given element.

STANDARDS FOR DATA BASE CONSTRUCTION

1. Each Physical File Record =
   - Control Segment (up to 30 characters)
   - Data Segment(s) (up to 240 characters)
     - Master File can have 1 to 9 segments
     - Variable File can have 1 segment
     - Segment Name is "SEGn' (e.g. SEG1)
2. Master File
   - # of links to Variable Files = 1 to 20
   - First 6 digits SEG1 = Last Trans Date
3. Variable File
   - All updating done via a single Base Master File
   - Must have Record Code (for TOTAL)
   - Has Record Type first 2 digits SEG1
   - Can have additional secondary control following Record Type

TRANSACTION CODES

AA = Add Master Record

DD = Delete Master Record and any associated Variable Files

CR = Change element in Master or Variable Record by Replacement

-or-

Create new Variable Record if none exists and insert element value from this transaction.

CE = Change element in Master or Variable Record by Erasure (set to blanks or zeroes as appropriate)

-and-

Completely eliminate a Variable Record if this was its last data element remaining.

CA = Change element by numeric addition or subtraction.

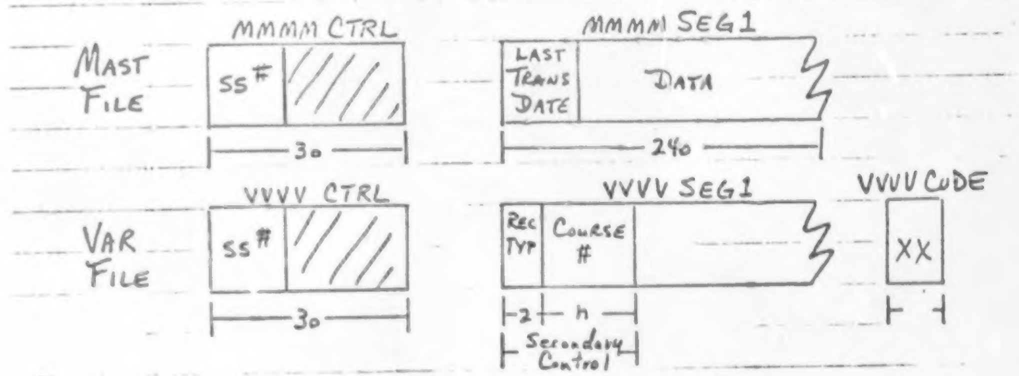


FIGURE 2

A direct change transaction uses one of the transaction codes shown in Figure 2. In effect, such transactions affect no file or item of data except that which is specified in the transaction itself. An example would be the recording of a student being registered in a specific course.

GM0030 can also call numerous other update modules for the purpose ofmaking indirect changes. Tailored programs, such modules are triggered by the type of transaction entered into the job stream. Thus, in any given run, many different types of files, activity, and indirect changes can be involved. An example of an indirect change could be the setting of an Enrollment Status Indicator, in a different file, when a student enrolls for any course.

GM0040 prints all error, unposted transactions that entered a given run, while GM0050 prints the posted conditions. In addition, GM0050 analyzes the users of data posted and generates, to multiple users if necessary, monitor reports of records that have changed.

Parameter-driven GRS

At any time, and at regularly scheduled reports, the GRS capability permits extraction of records from one or more Data Base files with or without sorting, for the printing of reports (lists, labels, accounting reports, statistics, etc.) all by parameter entry. It is expected that at least 80% of the computer center's output will be handled by the GRS -- if the Amherst installation is a good example.

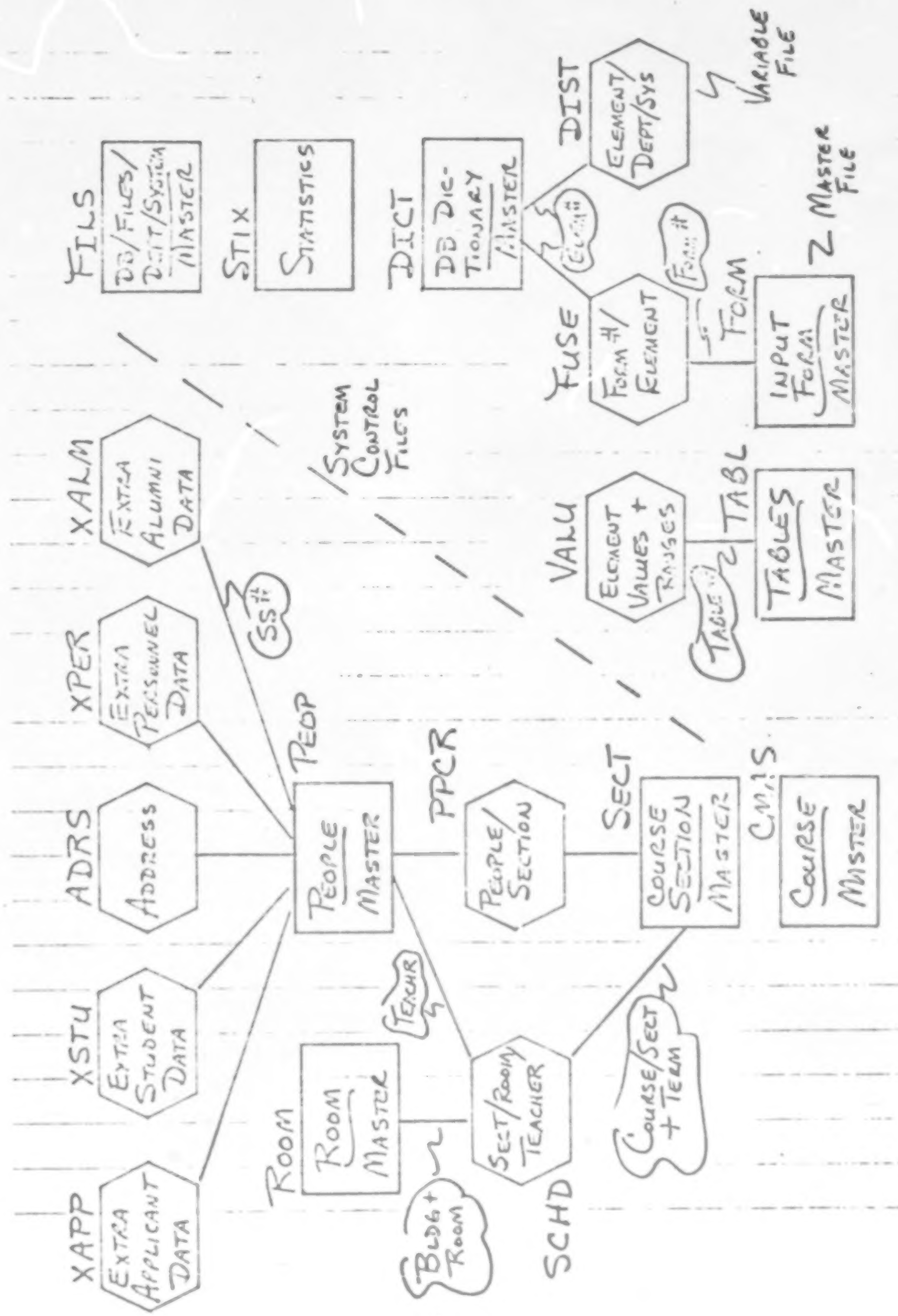
Data Base Standards of Construction

Constraints on construction of the DePaul Data Base were established in order to use the GMS to maintain all files with but one update module. First, some definitions are in order.

* Physical File = Grouping of similar records within the Data Base. TOTAL permits two types, master or single-entry and variable or variable-entry. Each is identified to TOTAL by a four-letter name, such as PEOP (for a PEOPLE file of names, etc.).

* File Record = Logically unique data grouping for a given primary control (like Soc. Sec. #) in the case of a master file. A variable file record needs both a primary control segment and a secondary control (part of SEG1) to establish its uniqueness -- and thus GMS can find the record when trying to update it.

* Segment = Grouping of data elements within a file record. CTRL segment contains only the primary control element.


XAPP
Extra Applicant Data
XSTU
Extra Student Data
ADRS
Address
XPER
Extra Personnel Data
XALM
Extra Alumni Data
FILS
DB/Files/Dept/System Master
STIX
Statistics
DICT
DB Dictionary Master
DIST
Element/Dept/Sys
Variable File
Z Master File
PEOP
People Master
SS#
System Control Files
ROOM
Room Master
Bldg + Room
Teachr
SCHD
Sect/Room/Teacher
Course/Sect + Term
PPCR
People/Section
SECT
Course Section Master
CMIS
Course Master
VALU
Element Values + Ranges
Table #
TABL
Tables Master
FUSE
Form #/Element
Elem #
Form #
FORM
Input Form Master

DePaul Data Base

Figure 3

A data segment, i.e. SEG1, can contain any number of data elements.

* Data Element = Usually the smallest data component of a segment. System permits multiple definition of elements.

While the GMS programs could be modified to accommodate almost any type of Data Base system, the setting of standards within our own has greatly facilitated the development of new methods and techniques for dealing with the data contained therein.

Data Base Dictionary of Elements

Several files in the Data Base are used solely for control of the system. Perhaps the most important is the DICT or Dictionary of Data Elements File. Every element is described in DICT, its associated file, segment, beginning position, size, class, etc. Each element has a unique number and is structured in a special way as seen in Figure 2. The element in the DePaul system is related to a physical file by its prefix.

In the final analysis, the DICT even describes itself, as seen in Figure 4. Here, it can be seen that 22 elements comprise one DICT record, 21 of which are in SEG1 and 1 in CTRL segment. A 23rd element really exists in DICTSEG1 positions 1 - 6 known as "last transaction date" which is automatically supplied and maintained by the GMS. Because the DICT can describe itself, the GMS can be used to update the DICT records of any element. In other words, changes to the structure of the system are as simple as changes to the data.

Data Base File Relationships

Another system control file FILS describes the characteristics of each physical file and its relationship to others in the Data Base; see Figure 3 for a sample Data Base Schematic and Figure 4 for the associated content of the FILS file. An interesting fact is that FILS can exist under any data base access method (not necessarily TOTAL). File relationships are also carried in the TOTAL core-resident descriptor module, but that is frequently proprietary, changeable or the vendor may even be changed. Like other data base files, FILS can be modified through GMS. FILS and DICT along with the attendant standards of data base construct, terminology and file relationships, form the heart of the DePaul system. Other system control files are explained later.

```
INITIAL LOAD OF DATA BASE    FILS    DATA ELEMENTS          09/01/72

NO.     M/V SGS LKS UPDT LK   LK01 LK02....ETC.
......   .   .  .. ....  ..   .... .... ....
FILS     M   1  00        0
DICT     M   1  00        0
PEOP     M   2  02        0   PPCR ADRS1
COUR     M   1  01        0   PPCR
PPCR     V   1   0 PEOP  01
ADRS     V   1   0 PEOP  02
MF01         0  04        0   FILS DICT PEOP COUR
VF01         0  02        0   PPCR ADRS
PE0114       B   2  007020  027003
PC0114       B   2  007020  027003
AD0114       B   2  007020  027003
PE0115       A   3  007012  030004  027003
CR0114       A   1  007015
```

```
INITIAL LOAD OF DATA BASE DICTIONARY DATA ELEMENTS          09/01/72

NC.   M/V RCOD FILE/SEG SIZE LOC CLS DEC CHG   NAME    DESCRIPTION                DEPT R SHIFT UPC
.....  .   ..  ........ ...  ...  .   .  ...  ......   ........................   ... .  ...   ...
DI001  M       DICTCTRL   5    1  X             DINUMB  DATA ELEMENT NUMBER        114 C  000
DI020  M       DICTSEG1   1    7  X         ER  DITYPE  FILE TYPE                  114    000
DI030  M       DICTSEG1   2    9  X         ER  DIRCOD  RECORD CODE - TOTAL        114    000
DI040  M       DICTSEG1   8   10  X         ER  DISEGM  SEGMENT NAME / FILE NAME   114    000
DI050  M       DICTSEG1   3   18  N   0    ERA  DILENG  ELEMENT LENGTH %BYTES¤     114    000
DI060  M       DICTSEG1   3   21  N   0    ERA  DILOCA  ELEMENT LOCATION %LEFT¤    114    000
DI070  M       DICTSEG1   1   24  X         ER  DICLAS  ELEMENT CLASS              114    000
DI080  M       DICTSEG1   1   25  N   0    ERA  DIDECM  DECIMAL PRECISION          114    000
DI090  M       DICTSEG1   3   26  X         ER  DICHNG  VALID CHANGE TYPES         114    000
DI100  M       DICTSEG1   6   29  X         ER  DIMNEM  ELEMENT MNEMONIC           114    000
DI110  M       DICTSEG1  25   35  X         ER  DIDESC  ELEMENT DESCRIPTION        114    000
DI120  M       DICTSEG1   3   60  N   0    ERA  DIDEPT  MAINTENANCE DEPARTMENT     114    000
DI130  M       DICTSEG1   1   63  X         ER  DIREST  CHANGE RESTRICTION CODE    114    000
DI140  M       DICTSEG1   3   64  N   0    ERA  DISHIF  SHIFT LOC %AUTOM¤          114    000
DI150  M       DICTSEG1   3   67  N   0    ERA  DIUPDT  UPDATE ROUTINE             114    000
DI160  M       DICTSEG1   2   70  X         ER  DITABL  VALIDATION TABLE NO.       114    000
DI170  M       DICTSEG1   2   72  X         ER  DIRTYP  RECORD TYPE - CODED REC.   114    000
DI180  M       DICTSEG1   3   74  N   0    ERA  DISLOC  SECONDARY CONTROL LOC. LF  114    000
DI190  M       DICTSEG1   2   77  N   0    ERA  DISLEN  SECONDARY CONTROL LENGTH   114    000
DI200  M       DICTSEG1   1   79  X         ER  DIOCCR  MULTIPLE OCCURENCE CODE    114    000
DI210  M       DICTSEG1   1   80  X         ER  DIESSN  ESSENTIAL/NON-ESSENTIAL    114    000
DI220  M       DICTSEG1   1   81  X         ER  DIMONT  MONITOR CARD CODE          114    000
```

```
INITIAL LOAD OF DATA BASE DICTIONARY DATA ELEMENTS          09/01/72

NO.   M/V RCOD FILE/SEG SIZE LOC CLS DEC CHG   NAME    DESCRIPTION                DEPT R SHIFT UPC
.....  .   ..  ........ ...  ...  .   .  ...  ......   ........................   ... .  ...   ...
PE001  M       PEOPCTRL   9    1  N   0         PECTRL  PEOPLE FILE CONTROL NO.    114 C  000   900
PE010  M       PEOPSEG1  12    7  X         ER  PELNME  LAST NAME                  114    000   900
PE020  M       PEOPSEG1   8   19  X         ER  PEFNME  FIRST NAME                 114    000   900
PE030  M       PEOPSEG1   3   27  X         ER  PECLAS  CLASS                      114    000   900
PE040  M       PEOPSEG1   4   30  X         ER  PEMAJR  MAJOR                      115    000   900
PE050  M       PEOPSEG2   1    9  N   0    ERA  PES01   ALUMNI STATUS              114    000   900
PE060  M       PEOPSEG2   1   10  N   0    ERA  PES02   ADMISSION STATUS           114    000   900
PE070  M       PEOPSEG2   1   11  N   0    ERA  PES03   ACTIVE STUDENT STATUS      114    012   900
PE080  M       PEOPSEG2   1   12  N   0    ERA  PES03P  STUDENT STATUS-PREV        114    000   900
PE090  M       PEOPSEG2   3   13  X         ER  PEGEOG  GEOGRAPHIC CODE            114    000   900
PE100  M       PEOPSEG2  10   16  X         ER  PESTRT  STREET ADDR - LOCAL        114    000   900
PE110  M       PEOPSEG2   8   26  X         ER  PECITY  CITY - LOCAL               114    000   900
PE120  M       PEOPSEG2   2   34  X         ER  PESTAT  STATE - LOCAL              114    000   900
PE130  M       PEOPSEG2   5   36  X         ER  PEZIP   ZIP CODE - LOCAL           114    000   900
```

FIGURE 4

Hardware Considerations

The system is designed to occupy no more than 48K of memory for any program module. For instance, the update phase GM0030 can include in memory at one time:

- Root Module GM0030, GM1030, Macros
- TOTAL Data Base Description Module (Size varies)
- TOTAL Data Base Manager
- Transient COBOL modules like XX0900

A 65K IBM 360/30 easily accommodates this system at a very modest hardware cost. Programming language for all primary modules is IBM COBOL D. There are also 3 macros written in Assembler, each about 15 instructions. The system is thus highly hardware and software-independent. Operating system is DOS.

Our double-density ITEL disk drives are able to provide concurrent operations among (1) 1401 Emulation, (2) Data Base Batch Processing, (3) Data Base TeleProcessing, and (4) Spooling of I/O operations. See Figure 5 for hardware maps.

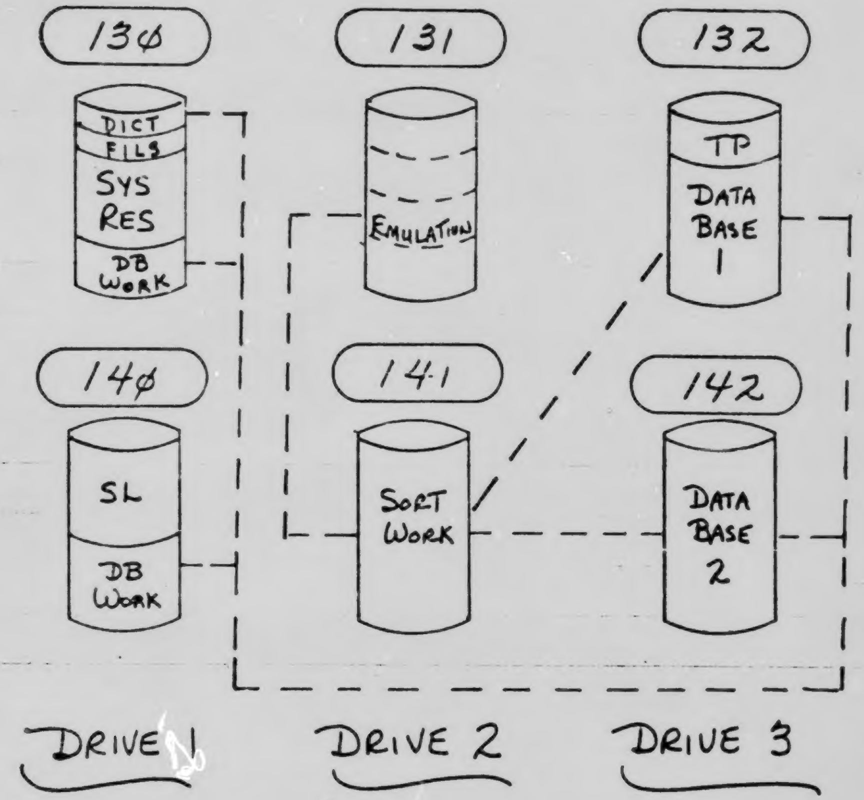
DATA BASE INFORMATION RETRIEVAL - GRS

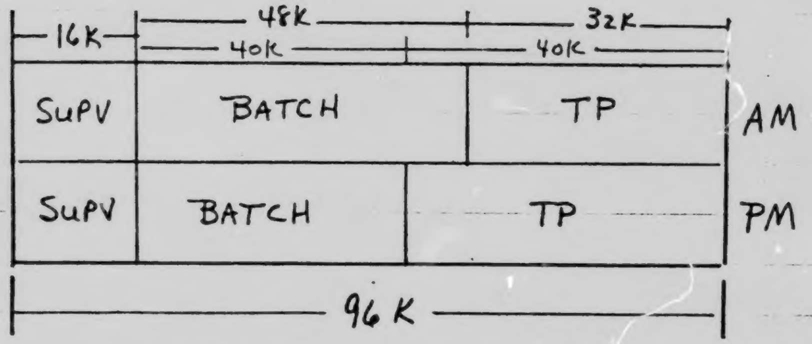
System Components

The Generalized Retrieval (and Reporting) System (GRS) basically follows the functional sequence of processing steps in Figure 6 which shows the GRS Job Stream. Components of this include:

- Generalized Extract Program - Written in COBOL, this is the key to combining information from several data base files into one, or more, meaningful logical system outputs.
- Generalized Sort Program - Nothing more than the IBM Sort Utility, the first 60 characters of each record produced by the Extract are sequenced. This is the key to successful multiple-report processing because it places all records for a given report together, each with its own independent internal sequence (such as name within college)...according to whatever data fields the Extract placed in the 60-character sort key.
- Generalized Print Program - An infinite variety of printed outputs is possible from this, our collection of program modules known as General Print.

DISK LAYOUT


130
131
132
DICT
FILS
SYS RES
DB WORK
EMULATION
TP
DATA BASE 1
140
141
142
SL
DB WORK
SORT WORK
DATA BASE 2
DRIVE 1
DRIVE 2
DRIVE 3


16K
48K
32K
40K
40K
SUPV
BATCH
TP
AM
SUPV
BATCH
TP
PM
96 K

MEMORY LAYOUT

FIGURE 5

GRS JOB STREAM

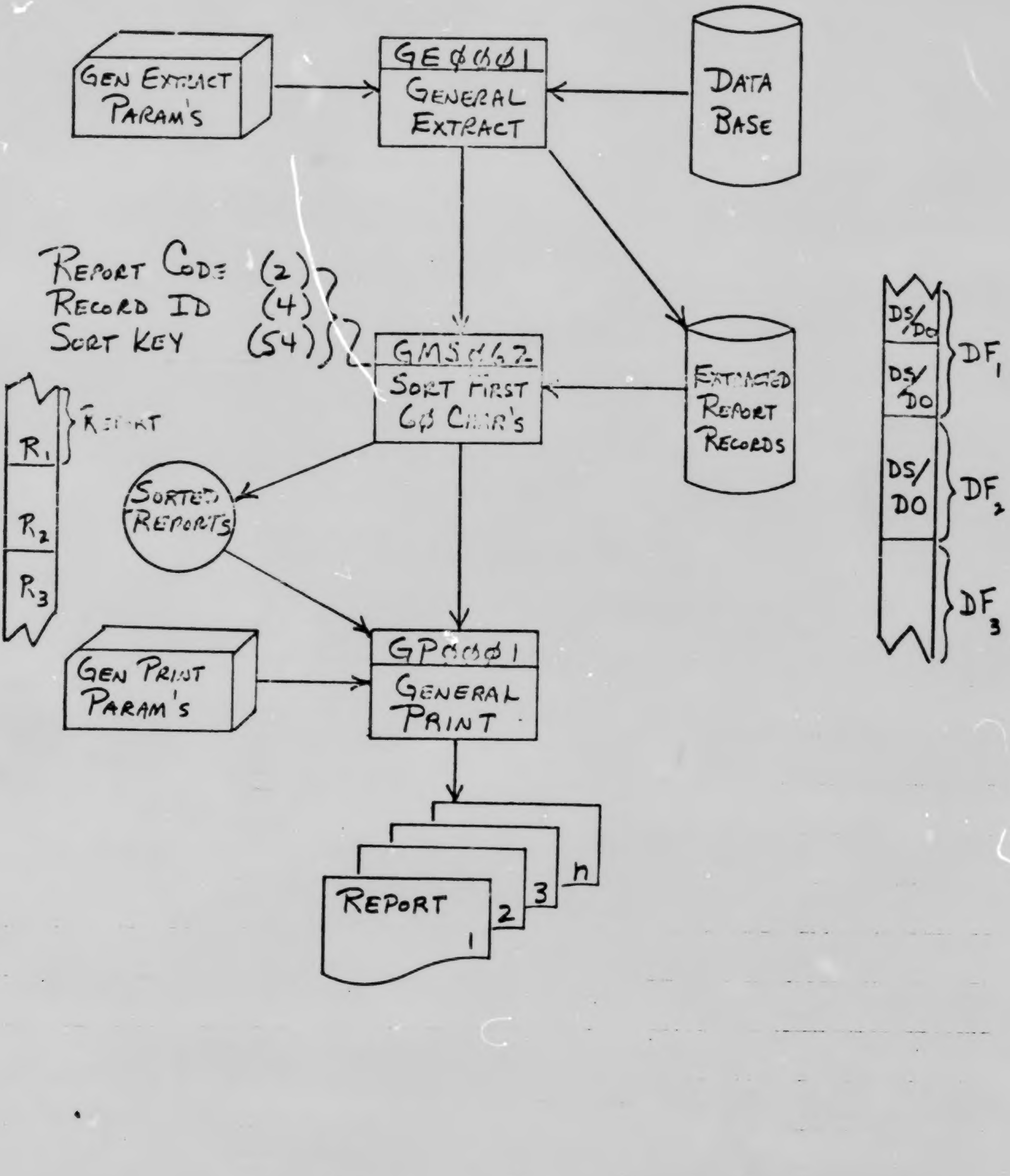

GEN EXTRACT PARAM'S
GE0001
GENERAL EXTRACT
DATA BASE
REPORT CODE (2)
RECORD ID (4)
SORT KEY (54)
GMS0062
SORT FIRST 60 CHAR'S
EXTRACTED REPORT RECORDS
REPORT
R1
R2
R3
SORTED REPORTS
DS/DO
DS/DO
DS/DO
DF1
DF2
DF3
GEN PRINT PARAM'S
GP0001
GENERAL PRINT
REPORT
1
2
3
n

FIGURE 6

System Characteristics

The GRS Job Stream can be used at any time to pass the desired data base files, extract data elements, sort and subsequently print them in any number of reports in any desired sequence and format. This, then, is the normal method for printing standard scheduled reports or other documents not associated with transactional activity to the data base. Special reports of an unscheduled nature can likewise be handled through this job stream.

The last phase of the GMS Job Stream is Monitor Record Generation, or the producing of specific records on which changes have taken place during the maintenance process. Monitoring is how a user knows what his computer data file looks like. It is as though the user had a "window" into his file which flicks the correct image of every record at him as he needs it.

Both reports and monitor records are produced by the system components Generalized Extract, Sort and Print. It was mentioned earlier that all extraction and printing is accomplished by preparing and entering parameter cards (also known as control cards) to specify the details of what is to be processed and how it is to be sorted and printed. An example of some Extract parameters is shown in Figure 8. Such parameters are:

* Fast and easy to prepare, but are oriented to the computer analyst, not the user.

* Optionally stored within the data base itself and retrieved by the GRS as needed to make data file extractions or prints.

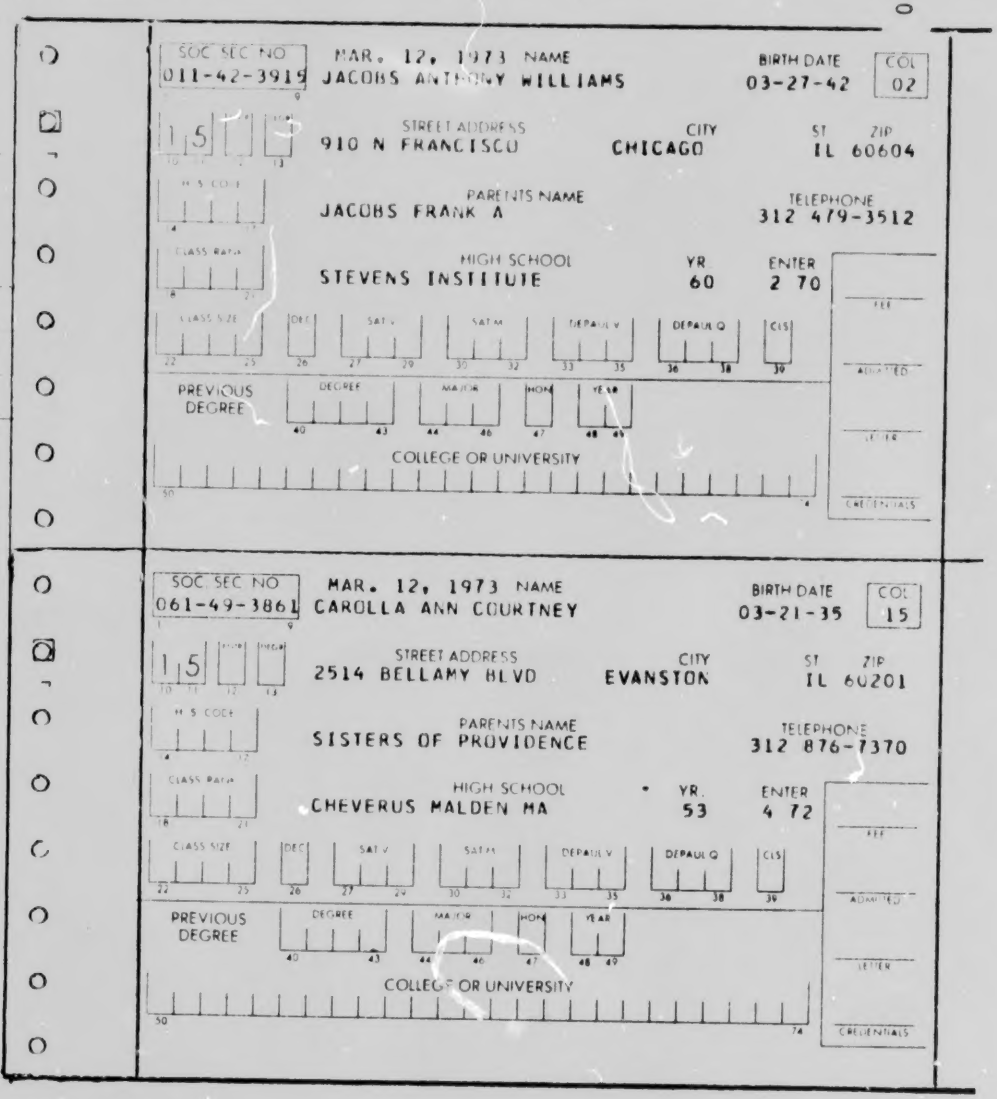
* Operational by default. COBOL own-code program modules can be written to over-ride the generalized functions of the system, through own-code exits provided in GRS.

Efficient computer centers have known for years that, although difficult to program and implement, extracting many reports from a data file at one time, then sorting and printing them in one solid stream (more or less automatically) was inherently better than processing reports one at a time. The savings can be attributed to less job set-up time, better use of computer resources (like the printer), fewer opportunities for manual intervention and error, etc. GRS adds to multi-report capability the following features:

* Any number of reports, up to 40, can be processed in a single run of the job stream.

* Reports on the Generalized Print input tape can be printed or bypassed via a Report Selection Parameter... making possible any manner of re-starts, re-runs, or repeat prints.

```
                DEPAUL PRINT GENERATOR          EDIT/ERROR LIST          MAR. 12, 1973          PAGE 0001
0FIXED                                  ADMISSIONS MONITOR MATRICULATION CARD
1N00500004
2000000000008
200100130A   X
3N0N00010C      001003001       SOC SEC     (LINE 1)
3N0N00020D      004-            EDITED      (LINE 1)
3N0N00030C      005002004                   (LINE 1)
3N0N00040D      007-                        (LINE 1)
3N0N00050C      008004006                   (LINE 1)
3N0N00060C      013030010       NAME        (LINE 1)
3N0N00070C      045002044       BIRTHDAY    (LINE 1)
3N0N00080D      047-            EDITED      (LINE 1)
3N0N00090C      048002046                   (LINE 1)
3N0N00100D      050-                        (LINE 1)
3N0N00110C      051002048                   (LINE 1)
3N0N00120C10    056002050       COLLEGE     (LINE 1)
3N0N00200C      013020079       STREET      (LINE2)
3N0N00210C      035014099       CITY        (LINE2)
3N0N00220C      050002113       STATE       (LINE2)
3N0N00230C12    053005115       ZIP         (LINE2)
3N0N00250C      013030150       PARENT      (LINE 3)
3N0N00260C      046003120M      PHONE       (LINE 3)
3N0N00270C      050003123       EDITED      (LINE 3)
3N0N00280D      053-                        (LINE 3)
3N0N00290C12    054004126                   (LINE 3)
3N0N00300C      013028226       HI SCHOOL   (LINE 4)
3N0N00310C      041002254       HS YEAR     (LINE 4)
3N0N00320C      047001057       ENTRY TRM   (LINE 4)
3N0N00330C12    049002058       ENTRY YR    (LINE 4)
```



SOC SEC NO 011-42-3919
MAR. 12, 1973 NAME JACOBS ANTHONY WILLIAMS
BIRTH DATE 03-27-42
COL 02
1 5
STREET ADDRESS 910 N FRANCISCO
CITY CHICAGO
ST IL ZIP 60604
PARENTS NAME JACOBS FRANK A
TELEPHONE 312 479-3512
HIGH SCHOOL STEVENS INSTITUTE
YR 60
ENTER 2 70

SOC SEC NO 061-49-3861
MAR. 12, 1973 NAME CAROLLA ANN COURTNEY
BIRTH DATE 03-21-35
COL 15
1 5
STREET ADDRESS 2514 BELLAMY BLVD
CITY EVANSTON
ST IL ZIP 60201
PARENTS NAME SISTERS OF PROVIDENCE
TELEPHONE 312 876-7370
HIGH SCHOOL CHEVERUS MALDEN MA
YR. 53
ENTER 4 72

FIGURE 7

* Reports to be printed are controlled against reports that were earlier extracted in an automatic sequence of console logging for operator and management use. A run summary is provided showing reports printed, bypassed, selected but missing, etc.

Simple extraction of all records from one or more data files does not constitute the retrieval of information. To be of any value, extensive record selection capabilities are necessary. Virtually any combination of selection criteria is permitted by both Generalized Extract and Print. In fact, records may even be selected by comparisons of several data fields within a given record, or by modification of certain data fields via table features explained below and then the result used for selection. In effect, selection is almost a miniature program written in parameter form and can be as simple or complex as one desires.

Either stock or custom pre-printed forms can be handled by GRS. A testline may be printed at the start of each report, as many times as necessary to align a special form...or optionally omitted, again, by report. Reports can be vertically and horizontally formatted in almost any fashion. Data from several records may be combined in a single print line or a single record may be spread vertically over any number of print lines...see sample Admissions Matriculation Card in Figure 7.

Complete editing of data fields, including designation of standard internal masks (or explicitly specified masks) and the emission of literals is provided. Headings can include automatic dating and paging by report. In addition, table cards may be entered by report for the expansion or conversion of data fields prior to printing. Control totals for numeric fields and record counts based on the record selection logic (up to 63 record counts per report) can be generated for accounting reports and/or statistical summaries.

System Operations - A Multiple Report Example

Figures 8 through 12 show the General Extract, General Print parameters and the resulting reports, respectively, for a multiple report run of the GRS Job Stream.

The DePaul Data Base being used is shown in Figure 3 depicting its various files. The sample run is intended to generate 4 reports from 3 separate sweeps of the data base by General Extract, each report with its own selection criteria, format, sequence and content.

General Extract parameters are coded as DF (Define File), DO (Define Output) or DS (Define Selection) in char. 1-2. The DF group repeats for each sweep of the data base. DF cards identify the File Extraction Path the program will follow in order to retrieve data fields from files. Data fields thus retrieved will be used for testing purposes (according to the DS cards) or for output specifications (according to DO cards). DS and DO parameters must be related to each other by the identification in chars. 3-7 (e.g. TEST01). More than one DS/DO set can be entered for a group of DF cards.

```
 // JOB GRJ001.P D114     GEN. REPORT - XTRACT TO DISK                              DATE 03/12/73,CLOCK 1
     ***********************************************************************************
692
 * BEGIN GEP001 - GENERALIZED XTRACT TO DISK
 *
 // OPTION DUMP
 // ASSGN SYS005,X'00C'    INPUT IS FROM CARDS
 // ASSGN SYS010,X'141'    XTRACT OUTPUT DISK--FORWARD TO GMS062
 // DLBL SYS010,'WORK D',0
 // EXTENT SYS010,111111,,,2020,800
 // EXEC GEP001

 DF0PEOPPE001              *1ST PASS*   GET RECORD FROM PEOP FILE
 DF1XTRAPE001PEOPLK04                   THEN A RELATED RECORD FROM XTRA FILE
 DF2ADRSPE001PEOPLK02                   LAST, A RELATED RECORD FROM ADRS FILE
 DOTEST01010 *(0001)AA0101*             (REPORT AA,FORMAT 0101) - PUT IN 1ST 6 CHAR
 DOTEST01020 ,XT216(0061),XT266(0065),XT316(0071)         COLLEGE
 DOTEST01030 ,PE001(0080),PE028(0100),AD308(0150)         SS,NAME # HI SCHL NAME
 DOTEST01040 ,PE028(0007)               SORT NAME - PLACE IN SORT KEY
 DSTEST01010 PE044EQAAABBB *F*          SELECT SEX F
 DOTEST02010 *(0001)AB0101*             REPORT AB
 DOTEST02020 ,PE028(0061),PE032(0100)                     NAME # BIRTHDAY
 DOTEST02025 ,PE208(0120),PE212(0135),PE216(0140)         CITY,STATE # ZIP
 DOTEST02030 ,PE216(0007),PE028(0015) SORT ZIP AND NAME
 DSTEST02010 PE044EQAAABBB *M*          SELECT SEX M
 )* ****  OPENM PEOP
 )* ****  OPENV XTRA
 )* ****  OPENV ADRS
 C000000 RECORDS NOT FOUND
 C000113 PEOP       RECORDS EXTRACTED
 C000109 XTRA       RECORDS EXTRACTED
 C000052 ADRS       RECORDS EXTRACTED
 C000109 DISK RECORDS WRITTEN
 )* ****  CLOSM PEOP
 )* ****  CLOSV XTRA
 )* ****  CLOSV ADRS

 DF0FUSEFS001              *2ND PASS*   GET RECORD FROM FUSE FILE
 DF1DICTFS030                           THEN GET RELATED RECORD FROM DICT FILE
 DF2FILSDI040                           LAST, GET RELATED RECORD FROM FILS FILE
 DOTEST03010 *(0001)XX0101*             REPORT XX
 DOTEST03020 ,FS001(0061),FS030(0070)                     FORM # ELEMENT NO.
 DOTEST03025 ,DI050(0080),FL010(0090)                     ELEM LENGTH # FILE TYPE
 DOTEST03030 ,FS001(0007)               SORT FORM NO.
 DSTEST03010 FL010NEAAABBB *M*          SELECT ONLY MASTER FILE ELEMENTS
 )* ****  OPENV FUSE
 )* OPEN  OPENM DICT
 )* OPEN  OPENM FILS
 C000000 RECORDS NOT FOUND
 C000146 FUSE       RECORDS EXTRACTED
 C000146 DICT       RECORDS EXTRACTED
 C000000 FILS       RECORDS EXTRACTED
 C000146 DISK RECORDS WRITTEN
 )* ****  CLOSV FUSE
 )* ****  CLOSM DICT
 )* ****  CLOSM FILS

 DF0VALUVA001              *3RD PASS*   GET RECORD FROM VALU FILE
 DF1TABLVA001                           LAST, GET RELATED RECORD FROM TABL FILE
 DOTEST04010 *(0001)YY0101*             REPORT YY
 DOTEST04020 ,TB010(0061),VA020(0100),TB001(0200)         TABLE NAME, VALUE # NO.
 DOTEST04030 ,TB001(0007)               SORT TABLE NO.
 )* ****  OPENV VALU
 )* ****  OPENM TABL
 C000000 RECORDS NOT FOUND
 C000047 VALU       RECORDS EXTRACTED
 C000047 TABL       RECORDS EXTRACTED
 C000047 DISK RECORDS WRITTEN
 )* ****  CLOSV VALU
 )* ****  CLOSM TABL
```

FIGURE 8

Two reports, AA and AB, are being generated from the first data base pass by General Extract. Selection of females for the first report and males for the second is reflected in the respective DS cards. Note the data fields to be sorted are specified in the DO cards (such as in TEST01) where name element PE028 is being placed in position 0007 of the output record following the literal 'AA0101'.

Note that a data pass by the extract program involves only those data files specified in the DF cards, no others. Thus, extraction speed is limited to the time necessary to pass the disk files from beginning to end (sequential read). Multiple reports (DS/DO's) can, of course, be generated during that pass.

For cases where the specific records to be retrieved can be identified by their control number (like Soc Sec #), the extract program reads a "trigger" file of cards or tape containing the specific control numbers desired. A direct seek is then made to the records wanted and extraction is accomplished immediately. This is the technique used to generate Monitor Records from the GMS Job Stream each day.

After the sort, General Print accepts the sorted input tape along with a deck of print parameter cards, in report code sequence (e.g., AA,AB,XX,YY). These parameter cards are identified by char. 1 which can range from 1 to 6 as follows:

* Param 1 - Provides report code in chars. 2-3 along with input record length and blocking factor.

* Param 2 - Heading information. Can also contain standard line spacing and several other options for the report.

* Param 3 - Detail print line information including designators as to when line is to print and spacing before or after print. This card is also used to specify a Summary report line...not included in example.

* Param 4 - Table cards for Data Field Expansion/Conversion feature.

* Param 5 - Record Selection information. Although not shown in example, this capability is similar to General Extract feature.

* Param 6 - Control Break, Summarizing and Crossfooting specifications.

General Print matches the report records on the input tape to the parameter cards by report code (AA on beginning of tape, for example). When a Param 1 card report code is found on the tape, the rest of the parameter cards for the report are read in and analyzed (stops reading cards when next Param 1 card is read) and report printing begins.

```
            DEPAUL PRINT GENERATOR         EDIT/ERROR LIST          MAR. 12, 1973        PAGE 0001
0FIXED
1AA0500004                            REPORT AA
2000000000045
20010001001PEOPLE REPORT 1
20010040001 X
20010060001PAGE 0000
301010010C    001009080               AA - SOC/SEC
301010020C    015030100               AA - NAME
301010040C    050002061               AA - COLLEGE UG
301010050C    055002065               AA - COLLEGE GR
301010060C    060002071               AA - COLLEGE PR
301010070C12  070025150               AA - HS NAME
1AB0500004                            REPORT AB

TESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTLINETESTL

PEOPLE REPORT 1              MAR. 12, 1973                PAGE    1

031147446       ABRAM,ALICE DOUCET             09

024055501       ARNDT,MARGARET T                   15

027424329       BLAKESLEE,BETSEY SYL               11

025229799       BOU-SLIMAN,JOAN BARB               11

040483614       BRYANT,SARAH M                 01

028142279       CARSON,MARY SILVANO            09

034424363       CHANG,THERESA                  07

009220495       CORMIER,LUCILLE J              03
```

FIGURE 9

```
                DEPAUL PRINT GENERATOR        EDIT/ERROR LIST        MAR. 12, 1973       PAGE 0002
200000000045
200100010001 X
200100030001PEOPLE REPORT 2
301010010C    001030061        AB - NAME
301010020C    035002100        AB - BIRTHDAY
3010100220    037/
301010024C    038002102
3010100260    040/
301010028C    041002104
301010030C12 050014120         AB - CITY
301010040C10 050002135         AB - STATE
301010050C10 050005140         AB - ZIP
1YY0500004                     REPORT YY
```

```
MAR. 12, 1973            PEOPLE REPORT 2

PCLONSKY,LESTER                  06/21/51       CHICAGO
                                                IL
                                                60614

RYAN,THOMAS EDMUND               04730/47       CHICAGO
                                                IL
                                                60614

SANTIAGO,MIGUEL                  09/26/52       CHICAGO
                                                IL
                                                60614

VAILLANCOURT,DANIEL              03/05/47       CHICAGO
                                                IL
                                                60614

WCOD,RICHARD J                   02/26/11       CHICAGO
                                                IL
                                                60614
```

FIGURE 10

```
          DEPAUL PRINT GENERATOR        EDIT/ERROR LIST          MAR. 12, 1973       PAGE 0003
200000000045
200100050A TABLE REPORT
301010010C    00100320C         YY - TABLE NO.
301010020C    005025061         YY - TABLE NAME
301010030C11 035050100          YY - VALUE


     TABLE REPORT

G1   COLLEGE GRAD                     10111215

G2   CLASS LEVEL GRAD                 ABCDEFG

G3   MAJOR FIELD GRAD                 025070005007210227287320325140151152153154155161

G3   MAJOR FIELD GRAD                 385395400405410421422423424425460465470475505515

G3   MAJOR FIELD GRAD                 162180200220225235240255265290330335360362365370

G3   MAJOR FIELD GRAD                 550999

G4   DEGREE STATUS GRAD               ND

G5   ADMIT STATUS GRAD                123456R

L1   COLLEGE PRO                      1314

L2   CLASS LEVEL PRO                  1234

L3   MAJOR FIELD PRO                  297

L4   DEGREE STATUS PRO                ND

L5   ADMIT STATUS PRO                 123456R

P1   ETHNIC BACKGROUND CODE           123456

P2   CITIZENSHIP VISA CODES           1234

P4   SEX F OR M                       FM

P5   RELIGION  C OR N                 CN

P6   MARITAL STATUS  M OR S           MS

U1   COLLEGE UNDERGRAD                010305060708092122

U2   CLASS LEVEL UNDERGRAD            12345678
```

FIGURE 11

```
        DEPAUL PRINT GENERATOR        EDIT/ERROR LIST          MAR. 12, 1973       PAGE 0004
RSG PRINT AT END OF JOB. ALL REPORTS WERE SELECTED.
   3 REPORTS WERE PRINTED.
   0 REPORTS CONTAINED ERRORS.
   0 REPORTS WERE NOT PRINTED DUE TO CONTROL CARD ERRORS.
   0 REPORTS ON DATA FILE WERE NOT SELECTED.
   4 REPORTS ON DATA FILE.
   0 SELECTED REPORTS NOT ON DATA FILE.
   2 SELECTED REPORTS NOT ON CARD FILE.
--0000305 RECORDS ON DATA FILE.
  0000014 PAGES PRINTED.
```

CONSOLE LOG

```
BG  // JOB GP0001.T D114        GENERALIZED PRINT PRODUCTION
   DATE 03/12/73,CLOCK 13/57/31
BG  REPORT CODE    NOT INCLUDED ON CONTROL (CARD) FILE.
BG  REPORT CODE AA READY TO PRINT
BG  T=TESTLIN P=PROCEED
BG  T
BG  T=TESTLIN P=PROCEED
BG  P
BG  REPORT CODE AB READY TO PRINT
BG  T=TESTLIN P=PROCEED
BG  P
BG  REPORT CODE XX NOT INCLUDED ON CONTROL (CARD) FILE.
BG  REPORT CODE YY READY TO PRINT
BG  T=TESTLIN P=PROCEED
BG  P
BG  EOJ GP0001.T
   DATE 03/12/73,CLOCK 13/59/14,DURATION 00/01/43
BG
BG
```

FIGURE 12

ELEMENTS OF DATA DICTIONARY (CONTROLLER) SYSTEMS DESIGN

Paul J. Plourde
Director
Computer Center
Amherst College

The very fact that we have assembled here an audience of professionals listening to someone talk about Data Base Dictionaries is indicative of the current vogue among systems designers which is to jump onto what has now become a large bandwagon, namely utilization of a Data Base Management System.[1]

When one considers using a Data Base Management System, one of the primary questions that one must ask oneself is how to manage a Data Base Management System. While this may sound somewhat ludicrous, a closer investigation will reveal that most Data Base Management Systems lack some essential ingredients when it comes to providing certain vital functions that a working Data Base must provide. For this reason and others to be noted in more detail throughout this paper, the question of designing a Data Base Dictionary System, or perhaps more appropriately named,a Data Base Controller System, becomes vitally important to the success of the implementation of any system that utilizes a DBMS.

A discussion of design concepts for a Dictionary could encompass various aspects of computing ranging from merely a listing of elements stored (a directory) to a question of storing parameters for query and logic capability that, in effect, venture into the category of language development. It is not the express purpose of this paper to eliminate any of the possibilities within this wide range, but merely to discuss some of these possibilities.

Dictionaries as they have been designed by various leading users, have performed some or all of the following three functions:[2]

A. Description - this is the function which permits an installation to describe the physical characteristics and logical relationships of data stored in the Data Base.

B. Protection and Security - this encompasses the wide range of questions of who has access to the Data Base, in what ways can a specific data element be updated, and implies system safeguard capabilities, both internal to the computer and external, which may involve multiple information centers within an organization.

C. Facilitate Use - although there is some overlap between the prior two functions of a dictionary, this function addresses the broader question of language development, as well as merely the question of retrieval & update capability.

The preceding three functions service three different clientele who are associated with the use of a Data Base, namely the user (the owner/manager of the data), the systems designers and programmers, and the software itself. While inter-relationships between the dictionary and software is perhaps a more sophisticated potential use of a dictionary, uses of a Data Base Dictionary System should be a primary concern to systems designers and programmers and users should express equal concern even if they don't know specifically what such a system can provide them. At the very least, users should be aware of the data that's stored in the Data Base and those elements which they have the responsibility of maintaining. They should also be aware of the elements that they have access to and, if they have any degree of sophistication, most of the data stored in a typical dictionary (as outlined in more detail below) should be common knowledge to users.

The need for designing a dictionary for use by systems designers and programmers is not well documented and the best single reason for including a dictionary master file (as well as associated support files) in a[3] TOTAL Data Base is that it provides considerable flexibility to systems designers. Some elaboration is probably required here to explain this point and it is merely that if you provide enough filler space in your data segments defined in the Data Definition Language, and the dictionary acts as a sub-definition of the data segments, then by simply changing the dictionary through some file maintenance procedure, one can alter the effective data elements stored in the Data Base without having to regenerate the Data Base.

While defining a new Data Base may not seem to be a major problem for most of you, having the flexibility of modifying the elements stored in the Data Base simply by doing file maintenance to a dictionary file is far simpler than Data Base redefinition. It has been previously stated that dictionaries service three functions: to describe, protect/secure, and facilitate use of data in a Data Base. These three functions are exercised for four categories of data description as follows:

A. Data stored in the Data Base.

B. Data input to the Data Base, that is transaction-oriented data as opposed to merely storing an element in the Data Base.

C. Output specification (report descriptions and report elements).

D. Processing data - this is a description of a data that is not input nor processed through a trans-

> action, but exists for the purpose of processing, that is an element or groups of elements created for processing purposes but which do not exist as elements in and of themselves.

The design of a Data Base Dictionary System (Controller System) is typical of application systems design in that there is not one right way to proceed. When one considers the many approaches to this problem, we end up asking ourselves many more questions than we can answer at first glance and it forces us to make choices about the direction in which we wish to proceed.

The first and most obvious question is whether or not one wants to maintain a manual dictionary system or wishes to seek a programming solution to the problem. It should not be implied that the choice of a manual system will negate all of the problem areas listed below but mainly places them in a different perspective.

The first problem is the question of how many levels[4] of this system should one implement. Should the system include merely data stored in the Data Base or attempt to describe all input transactions or go to the extent of specifying output specifications or including data that is processed within this system as well.

The next question is how many levels of users, that is systems designers/programmers, users or software, should be serviced by the system that is designed.

The answers to these questions are not global but are localized and the system must be defined to satisfy the goals defined in your installation, but the most important factor is to have a plan and get started.

If one gets involved in the purist's view of a Data Base Dictionary System,[5] you might end up designing a system that has fifteen master files and twenty-two variable files, as was the case with one leading user. While such a system may be the most sophisticated thing yet known to man, it will not quickly satisfy any of the categories of data description nor will it service many levels of users in the very near future. This author would argue that a phased implementation of such a system is far better than committing oneself to an all out discussion of the feasibility of a comprehensive system at the outset.

The next major problem that a dictionary system is faced with is interfacing with the Data Base Definition or whatever that might be called with the particular Data Base System that one is using. The first question is how does one get a picture of the Data Base into the dictionary system, and the corollary question is how do you keep the Data Dictionary System and the Data Base Generated System in synchronization with each other. For users of the TOTAL DBMS[6] in particular, having a picture of the Data Base Generation[7] in the Data Dictionary System is most useful to identify links, record codes, file names, whether a record is multiply occuring or not, all of which are most useful if one is contemplating designing a system that recasts (transfers) all of the data associated with a given control information from one point in a Data Base to another. A second purpose would be to identify segment lists and the sizes of all segments and data elements for purposes of maintaining the Data Base in a generalized fashion (general maintenance).

Another problem and one which might sound as if we are begging the question is the endless problem of how one controls the controller. Once again, we have a choice of a manual system or whether this is handled through programming. There are a number of questions, which fall into the category of cross-referencing, concerning whether an entry in the Data Dictionary System (a) conforms to program usage of that element, (b) validates the data as it is being input in the Data Base, and (c) checks the effect on another part of the system when an element is changed in the Data Dictionary System. I have indicated that this is an endless problem because one could conceivably write a system that controls the Data Dictionary System. This is another situation where one has to decide where the system stops and human decision-making takes over but I would not dismiss the question of cross-reference checking within a Data Dictionary System out of hand as merely something that is a blue-sky approach to Data Management.

Another significant problem when one considers Data Management Systems in general and Data Dictionary Systems in particular is how to prevent system crash. The answer to this question reminds me somewhat of the reasoning behind utilization of a simulation model. You may not find out anything you didn't already know but at least it forces you to go through the exercise of gathering the data together and presenting it in a systematic form for analysis. In this case you have to ask yourself the question of when to allow changes to the Data Dictionary System, how should these changes be edited, how should they be tested, and whether or not maintaining a dictionary system is better than the alternative which is merely to write additional COBOL programs. There is also the question of when one modifies the Data Base itself (by doing a new Data Base Generation) and how one verifies that the dictionary is now in synchronization with the new generation (a problem noted above).

All of the preceding if not handled properly either by program or manual checking can result in a system crash when next one utilizes the Data Base and it is important to address these questions and document the procedure by which each should be handled if one is to be operating in a Data Base environment.

The final problem[8] that I would like to allude to is the whole question of documentation and whether or not this should be left for COBOL programs or whether the Data Dictionary System should assume a portion of this responsibility. While I cannot offer any specific solutions since those to be found are not global answers, I would merely raise the question in order that an installation consider it as they plan their system.

An all-encompassing Data Dictionary System would be one in which the whole system were self-documenting. It would probably fall in the category indicated earlier of having multiple masters and multiple variable files[9] and would perhaps require a full time person merely to maintain this system which in itself maintains the Data Base Management System. One such typical system is indicated in Attachment A and you can quickly see that there would be cross-referencing of data element utilization by program, application, file, segment, data name, utilization area, transaction type, and finally, table values. The significant question that one must ask oneself is how far down this road should your specific installation proceed.

Having dwelled for some time on general concepts of dictionary design, I would now like to turn to a discussion of some specific elements that might be stored in a dictionary system relating to two specific categories of data description, namely data stored in a Data Base and data input via transactions.

Listed on the following pages you will find some of the elements that individuals have chosen to store in dictionaries and an explanation of what this specific element is utilized for. It is important to note at the outset that you will not find any reference relating to terminal utilization of a Data Base Dictionary since time did not permit a review of such systems. There are some excellent systems in existence that address this question specifically & I would reference one such system, OASIS, which was developed at Stanford University, that has gone to extensive detail to insure terminal security and provide all of the facilities required of a dictionary when it is utilized in cooperation with an on-line query and update system.

MAINTENANCE DICTIONARY

D001 Data Element Number - This is the Control Field for a dictionary record. It must be a unique code.

D020 File Type

M = Total*Master
I = Index Sequential
S = Sequential (Serial)
T = Tape
V = Total*Variable
A = VIM (Acct)
D = Direct/Random
C = Card

D030 Record Code Identifier - indicates that the segment in which the element resides has a record code as defined by VVVVCode in the DBGEN.**

D040 Segment Name/File Name

TOTAL - This is the actual segment name, the first four bytes of which are file name.

D050 Element Length - This indicates the size of the element in bytes.

D060 Element Location - This indicates the leftmost position of the element (relative to 001) in the record or segment as stated in D040.

---

*In this case, Total refers to the Total Data Base Management System.
**DBGEN refers to the program through which the data base definition language is processed.

D070 Element Class - This specifies the class of the data:

A = Alphabetic
N = Numeric (Unsigned)
P = Packed Decimal
X = Alphanumeric
S = Numeric (Signed)

D080 Decimal Precision - This indicates the number of decimal places in a numeric element, must be completed if D070 equals N or S; if D070 is P, then this is for documentation only.

D090 Valid Change Types - This specifies the types of changes which can legitimately be made to this element.

E = Erase (set to blanks or zeros)
R = Replace
A = Arithmetic (only if D070 = N or S)

D100 Element Mnemonic - This Mnemonic will provide a unique identifier for the element and will be used as its standard COBOL label.

D110 Element Description - A brief description of the element oriented toward user identification.

D120 Maintenance Department - The department number of the department responsible for the maintenance of the element.

D130 Restriction Code - This code is used to indicate an element which has a special purpose.

C = Control Element - The element utilized to access the data segment, add (AA) or delete (DD) maintenance is valid while change maintenance is invalid.

E = Variable Erase - When the element is maintained this will cause all variable segments for the specified control to be deleted. D090 can only be "E".

I = Internal - The element is used by the system and should not be maintained.

D130 Restriction Code (cont.)

L = Linkage Element - The element utilized to access data in a master file when a linkage path exists between the element and the master, all linkage path data must be present when the variable record is initially added to the Data Base and cannot have change maintenance processed against its linkage elements.

N = Index - Specifies that the element is used to index another file. D040 must be "INDEXFIL".

X = Extract - Specifies that the element is for extract purposes only.

VARIABLE MAINTENANCE

D160 Primary Link - This is the two byte link indicating the link path from the parent master through which maintenance will be performed on a variable file.

D170 Record Type

Blank - The record in which this element resides does not include a record type code.

Non-Blank - The actual record type (AR or SA or BB, etc.) in a variable file;must complete D180 and D190.

D180 Secondary Control Location*- Leftmost position in a variable record (VV-IO) of the secondary control.

D190 Secondary Control Length - The length in bytes of the secondary control. Required if D180 completed.

D200 Occurrence Code - This identifies whether or not this element exists in a variable segment that is singly occuring or multiply occuring.

---

*Secondary control is that portion of a variable record less the master control which makes it unique.

D210 Essential/Non-Essential Code - This code determines whether or not a variable record will be constructed if not already on file upon presentation of a "CR" or "CA" transaction for an element.

D220 Replacement Document Flag - This code indicates whether or not change activity to an element would occasion the printing of a replacement document.

D240 Distribution Code - This code indicates whether or not change activity to an element would occasion the printing of additional documents that would notify departments which use the element but are not responsible for its maintenance that it has been altered.

TRANSACTION DICTIONARY

D001 Transaction Control Number - A number must be assigned for each transaction element (data field) as well as for blank fields. The first two characters of the transaction element number will be the same for all elements within a transaction. The third and fourth characters represent the starting location of this element within the transaction.

D600 Data Base Element Number - If this transaction element corresponds to an element in the dictionary, enter that dictionary number here. This can be used to identify a key field (master access number) or for cross-referencing purposes.

D605 Length - This size of the element in bytes (field length).

D610 Class

X = Alphanumeric (letter, blanks, numbers, punctuation)
A = Alphabetic (letters only)
N = Numeric (numbers only)

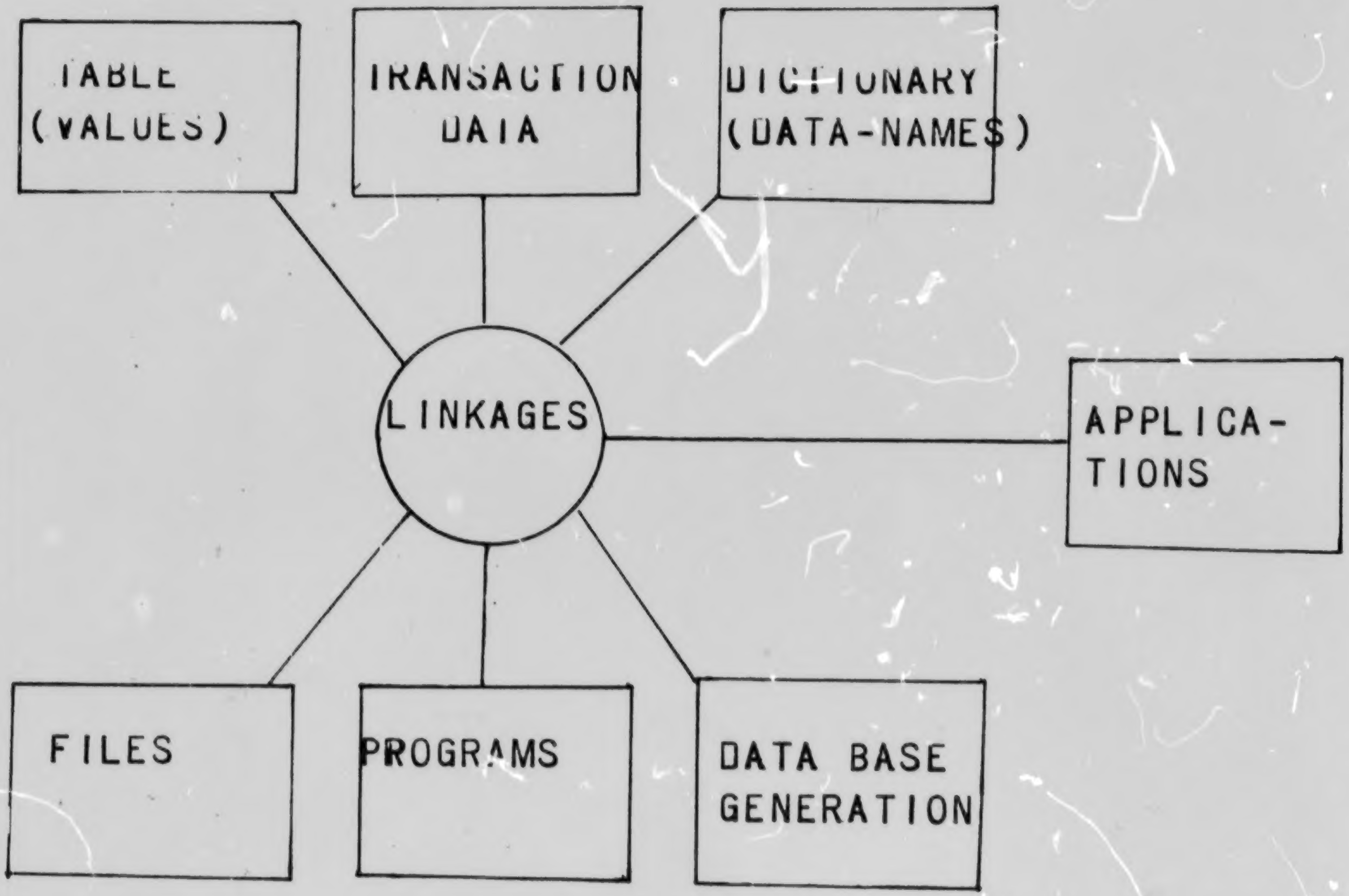


TABLE (VALUES)
TRANSACTION DATA
DICTIONARY (DATA-NAMES)
LINKAGES
APPLICA-TIONS
FILES
PROGRAMS
DATA BASE GENERATION

ATTACHMENT A

FOOTNOTES

[1] For an excellent introduction to data management definition and design see: An Introduction to Data Base Design, by John R. Lyon. New York: Wiley-Interscience, 1971.

[2] The GUIDE report on the "Data Base Administrator" identifies three similar functions as the responsibility of the data base administrator: namely, definition and organization, protection and documentation. Ref. ________. "The Data Base Administrator," New York: Guide, November 3, 1972.

[3] TOTAL in this paper refers to an all-encompassing data base that permits the integration of all major operating systems within an organizational entity (in this case a college or university). For a detailed review of this topic, see: TOTAL Systems Characteristics and Implementation, by C. C. Wendler. Cleveland: Systems and Procedures Association, 1966.

The design of such a system for a college or university is discussed in Plourde, Paul J. "TOTAL," Planning and Management Practices in Higher Education: Promise or Dilemma?, Denver, Colorado: Education Commission of the States, May, 1972, pp. 161-168.

An additional source that this author has found invaluable in defining and distinguishing TOTAL Systems, Integrated Systems and Management Information Systems is: Aron, J. D. "Information Systems in Perspective," Computing Surveys, Vol. 1, No. 4, (December 1969) pp. 213-236.

[4] The levels of design of a dictionary system should not be confused with the three levels of system design discussed in Aron, op.cit., pp. 217-218.

[5] Dodd, George G. "Elements of Data Management Systems," Computing Surveys, Vol. 1, No. 2, (June 1969), pp. 117-133 describes this and other essential components of a data management system.

[6] For an idex of the availability and characteristics of data base management systems (DBMS) see:

Koehr, G. J. and others. Data Management Systems Catalog. Bedford, Mass.: The MITRE Corp., January, 1973.

Anderson, Ruth K. Index of Data Management Software Packages. Washington: National Bureau of Standards Report, October, 1972.

________. "Trends in Data Management, Part 1," EDP Analyzer, Vol. 9, No. 5, May, 1971.

________. "Trends in Data Management, Part 2," EDP Analyzer, Vol. 9, No. 6, June, 1971.

[7] Data Base Generation will henceforth be used to refer to the output of the assembling of the Data Definition Language.

[8] While there is not one single reference that will outline all of the problems related to data base management, as a primer it would seem essential that the Codasyl Data Base Task Group Reports be read and that should be followed by the EDP Analyzer document "The Debate on Data Base Management."

________. Codasyl Data Base Task Group Report. New York: ACM, October, 1969 and April, 1971.

________. "The Debate on Data Base Management," EDP Analyzer, Vol. 10, No. 3, March 1972.

[9] A master file is a fixed size, single occurrence data set while a variable file is a variable size, multiply occurring data set.

BIBLIOGRAPHY

Anderson, Ruth K. *Index of Data Management Software Packages*. Washington: National Bureau of Standards, October, 1972.

Aron, J. D. "Information Systems in Perspective," *Computing Surveys*, Vol. 1, No. 4, December, 1969,.pp. 213-236.

__________. *Codasyl Data Base Task Group Report*. New York: ACM, October, 1969.

__________. *Codasyl Data Base Task Group Report*. New York: ACM, April, 1971.

Dodd, George G. "Elements of Data Management Systems," *Computing Surveys*, Vol.1, No. 2, June, 1969.

Koehr, G. J., and others. *Data Management Systems Catalog*. Bedford, Mass.: The MITRE Corp., January, 1973.

Lyon, John K. *An Introduction to Data Base Design*. New York: Wiley-Interscience, January, 1971.

Millard, R., and others. *Planning and Management Practices in Higher Education: Promise or Dilemma?* Denver, Colorado: Education Commission of the States, May, 1972.

__________. *The Data Base Administrator*. New York: Guide International Corp., November, 1972.

__________. "The Debate on Data Base Management," *EDP Analyzer*, Vol. 10, No.3, March, 1972.

__________. "Trends in Data Management, Part 1," *EDP Analyzer*, Vol. 9, No. 5, May, 1971.

__________. "Trends in Data Management, Part 2," *EDP Analyzer*, Vol. 9, No. 6, June, 1971.

Wendler, C. C. *Total Systems Characteristics and Implementation*. Cleveland, Ohio: Systems and Procedures Association, 1966.

AN ENVIRON/1 ADMINISTRATIVE ON-LINE SYSTEM

David Clements
Senior Programmer Analyst
Coast Community College District

AN ENVIRON/1 ADMINISTRATIVE ON-LINE SYSTEM

This presentation focused on the Administrative On-Line System now in use by the Coast Community College District. Discussion included a brief history of the history of the district's on-line experiences, the rationale for procurring a real-time software package (ENVIRON/1), and the philosophical guideline now employed for on-line applications development. The existing on-line inquiry and maintenance capabilities of the Student Information Application was described and graphically displayed in some detail. System logistical problems such as real-time data collection for 35,000 students per semester was approached. Abstracts of other on-line applications now in development were presented also. These included: Class Schedule Generation, District Facilities Inventory, a Transcript Micromation System, and a District Financial and Personnel System.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the author.

DATA ENTRY PRACTICE SURVEY PANEL

Harry Grothjahn
Director
Administrative Data Processing
University of Georgia

Frank Martin
Director
Data Processing
Michigan State University

Dennis Klinger
Associate Director
Southeast Regional Data
Center
Florida International Univ.

Wayne Ostendorf
Director
Administrative Data Processing
Iowa State University

DATA ENTRY PRACTICE SURVEY PANEL

Harry Grothjahn
Director
Administrative Data Processing
University of Georgia

Dennis Klinger
Associate Director
Southeast Regional Data Center
Florida International University

Frank Martin
Director
Data Processing
Michigan State University

Wayne Ostendorf
Director
Administrative Data Processing Center
Iowa State University

The general purpose of this session was to consider how changing key entry hardware and software can affect the data entry process. General items included in all the presentations were labor costs, input bottlenecks, accuracy, and the impact that input processing has in a data center.

Practical Considerations In Changing Key Entry Hardware was discussed by Wayne Ostendorf of Iowa State University. ISU's experience with various key entry systems including IBM 029's, 129's, UNIVAC 1710's, Mohawk 6401's (key tape) and IBM 3270's with the IBM Data-360 software package (key disk) was the main point of discussion in Ostendorf's presentation. Items considered included operator requirements, training, speed, scheduling, control, back-up of data, inserting records, deleting records, data editing, entry programs, etc. The advantages and disadvantages of punched cards, key tape and key disk systems were discussed. The general conclusion was that more productivity can be obtained from key tape and key disk than from key punched cards, however, all the previously mentioned items need to be studied before making massive changes.

An overview of the University of Georgia's Optical Scanning Study and Decision to acquire an optical scanner was presented by Harry Grothjahn. Key points in the discussion concerned choosing the criteria by which to evaluate optical scanning and that one may need to convert only a few applications to justify a minimum optical scanning system.

A review of the University of Florida South East Regional Data Center's Key Disk data entry was presented by Dennis Klinger. This center converted from key punched cards to key entry disk almost totally and reported improved production results as well as high satisfaction with the key disk concept.

Centralized vs Decentralized data entry at Michigan State University was discussed by Frank Martin. Pros and cons from an experience standpoint were considered. The general consensus was that providing support for both centralized and decentralized systems is the best alternative. The prime criteria considered were the application type and the user.

SOFTWARE PROCUREMENT PANEL

Robert C. Heterick
Director
Computing Center
Virginia Polytechnic Institute
and State University

Richard T. Paumen
Assistant Vice President
University Management Information
Systems
University of Pennsylvania

Ralph E. Lee
Director
Computer Center
University of Missouri
Rolla

Harry G. Vaughan
Director
Management Systems and
Analysis
Cornell University

SOFTWARE PROCUREMENT PANEL

Robert C. Heterick
Director
Computing Center
Virginia Polytechnic Institute
and State University

Richard T. Paumen
Assistant Vice President
University Management Information
Systems
University of Pennsylvania

Ralph E. Lee
Director
Computer Center
University of Missouri
Rolla

Henry G. Vaughan
Director
Management Systems and
Analysis
Cornell University

A panel of "Veterans" discussed the challenges and problems of procuring software to support the administrative processes of individual institutions of higher education. Topics under discussion ranged from the initial definition of requirements to software acceptance testing. Discussion included comments on related concerns including types of contracts, vendor evaluation procedures, negotiation techniques, contract terms, and subsequent changes. No written documentation of the panel discussion appears in the Proceedings.

The abstract of this paper is included for the interest of the reader. Unfortunately, the full paper was not received in time for publication. For more information, please contact the panel members.

THE USE OF APL AT A NEW AND EMERGING INSTITUTION

C. William Higginbotham
Assistant Director

Sampson G. Rice
Assistant Director

Management Information Systems
Governors State University

The Use of APL at a New and Emerging Institution

C. William Higginbotham, Ph.D.
Sampson G. Rice
Computing Center, Governors State University
Park Forest South, Illinois

Governors State University has joined with two other state institutions, Chicago State University and Northeastern Illinois University, in a cooperative effort to supply computing services to the three campuses. These services are provided by a Cooperative Computer Center, supported by the three schools, but funded as an independent institution. Access to the computer, an IBM 360/50, is achieved by two methods, remote batch entry using HASP/RJE and the APL time-sharing system. This paper describes some of our experiences using APL, decisions we have made and some of their consequences. Our reasons for selecting APL are presented, and a brief description of how it has been used since its installation is given. We also enumerate our more noteworthy successes and failures and some of our plans for future uses of the system.

BACKGROUND

Before proceeding, a brief history of GSU and the Cooperative Computer Center is in order. Governors State University was created by the Illinois legislature in a bill signed on July, 1969. It is under the jurisdiction of the Board of Governors of State Colleges and Universities, one of five governing bodies under the Board of Higher Education in the State of

Illinois. The university is located in Park Forest South, a new, planned community approximately thirty-five miles south of Chicago.

GSU is designed to serve urban minority students, primarily from low and middle income families. The school is a senior division university, admitting juniors, seniors and masters level graduate students and concentrating on service to junior college graduates. The average age of the students was originally twenty-seven or twenty-eight but has risen to thirty-two or thirty-three for the fall of 1973. The university is charged with becoming an innovative and experimenting institution, emphasizing job efficiency, functional citizenship, intra- and inter-personal relationships and cultural expansion.

In its first year of existence, 1969-70, the university had a nuclear staff of six professional and four support personnel. In the second year, it was decided to open the doors to students in the fall of 1971, a full two years ahead of the originally planned schedule. GSU opened in 1971 in temporary facilities located in a rented warehouse with 500 FTE students and a staff of forty professional and twenty support personnel. In 1972-73, the enrollment increased to 1200 and today our student body numbers about 1800 FTE. The student enrollment is currently increasing at the rate of 1000 FTE per year and it is projected to be 4000 by 1975 and 10,000 in 1980.

The administrative organization of the university is shown in Fig. 1. Reporting to the president are three vice-presidents: the Vice-President for Academic Affairs whose responsibilities include the four colleges, Admissions and Records (which includes the Registrar,) Financial Aids, and Student Services; the Administrative Vice-President who is responsible for the Personnel and Business Offices, Building and Plant Operations, and Safety Services; and the Vice-President for Research and Innovation whose responsibilities are institutional research, the Library and the Instructional Communications Center. Since Fig. 1 was created, the name MIS has been dropped in favor of the more general "Computing Center" and the unit now reports directly to the Office of the President.

There are two unique features of GSU which directly affect the administrative data processing function. The academic calendar is divided into six eight-week sessions corresponding roughly to the two-month periods September-October, November-December, January-February, March-April, May-June and July-August. This requires that the student register and be assessed for fees and tuition six times during the year. Furthermore, additions and deletions can be made at any time up to the sixth week of the session. This creates essentially a continuous registration process. The second feature relates to the learning modules (courses) which are competency based. No grades are given and the transcript includes a description of each competency which the student has successfully completed in a module. Thus, some of the transcripts may run to twenty or thirty pages of text which present a problem both in data entry and in large volumes of printing.

COOPERATIVE COMPUTER CENTER

Very early in the life of GSU, while we were still in the planning stages, the Board of Governors decided to attempt to decrease computing costs by centralizing computer facilities. Three of the five BOG schools were chosen to participate in the cooperative effort: Northeastern Illinois University (UNI,) Chicago State University (CSU,) and Governors State University (GSU.) These three were chosen because of their geographical proximity, all being in the greater Chicago area and because the computing services were not well developed at any of the three institutions. The initial situation was as follows: UNI was running an IBM 360 model 30 under DOS, CSU had an IBM 360 model 40 with DOS, and GSU was using a commercial APL time-sharing service.

In early 1971, the BOG Cooperative Computer Center was formed. It was separately funded by the BOG and the computing budgets of the three participating schools were drastically reduced. The CCC took over operation of the Model 40 at Chicago State and replaced it with a 360/50. The DOS operation was converted to OS with HASP in order to provide RJE capabilities to UNI. GSU recommended the installation of the APL time-sharing system to continue its use of this powerful and unique programming language.

REASONS FOR CHOOSING APL

Let us examine some of the reasons for GSU's choice of APL. The charge to the Computing Department, then known as

Management Information Systems, or MIS, was to support both the academic and administrative computing needs of the university. The most efficient way of providing academic computing services to a large number of faculty and students is through the use of a time-sharing system with many remote terminals for access. The APL language is an excellent tool for use in this mode in the classroom. It is an easy language to learn but yet very powerful. Students can start programming simple tasks very quickly and as they gain experience, they can progress to very complex programs written in a very few lines. Some other interactive languages are easy to learn but the programmer finds himself limited very quickly by the simplicity of the language and finds it necessary to learn other languages for the coding of more complex programs.

Many universities were beginning to use APL for their instructional computing needs. Orange Coast Community College has a very well known project under way for providing computer aided instruction through APL. The language was also being used extensively at Syracuse University and at the Universite Laval in Quebec, Canada.[1] There was a user's group in existence which provided the opportunity for exchange of programs, a very important consideration for an institution just starting its computing operation.

---

[1] See "Administrative Applications of APL" moderated by Adin Falkoff in these proceedings, Sessions P2.1 and P3.1.

In addition to the academic potential of APL, we foresaw that it would be a very valuable asset in the development of the administrative data processing function at GSU. Its matrix handling capability lends itself very nicely to report generation and formatting of output. An information system for management uses was developed at Xerox in APL and was so successful that it was decided to implement APL on the Xerox line of computers. Another recent very successful information system developed in APL is the American Airelines Information Management System (AAIMS.) APL would also be a very good language for use in implementing management forecasting systems such as RRPM. Another advantage of installing APL at the CCC was that it was immediately available. Both a DOS and an OS version existed and had been field tested for a number of years. It was also fully supported by IBM, a necessary condition since the small initial staff at CCC could not devote extensive resources to maintaining a time-sharing system. Also, APL required only a small initial investment in equipment. A number of selectric typewriter terminals and a teleprocessing line controller at the central computer site was all that was needed to provide computing services to the entire campus.

An interesting aspect of the decision to use a time-sharing approach to the provision of computing services, not only to the instructional units of the university but also to the administrative units, is that the user becomes very deeply involved in the whole process. This is nothing new in the areas of research and instruction where it is very common for faculty members and students to do all their own programming for computer related projects. On the administrative side, user control is relatively new. Traditionally, the administrative data processing function is highly centralized. A programming staff is usually maintained at the computer center which interacts with the user in specifying his requirements. The computing center staff then designs and implements the system and maintains control over data entry and report generation. If the user wishes to change the format of a report or change the data entry procedure to provide more up-to-date information on the files, it usually requires a major effort. We feel that our approach of user involvement in the data entry function as well as the scheduling of report production gets the user involved in the computing process in a positive way.

STUDENT RECORDS SYSTEM

To illustrate the results of our approach to administrative data processing, let us consider, in some detail, the development of our student records system. Initially, APL imposed some severe restrictions on the amount of data we could handle. The time-shared workspace was 48K bytes of memory into which the program and any data it might reference had to fit. However, some generalized data handling routines were written and a student file was established which consisted of thirteen data elements for each of 500 to 700 students. Since the file was completely on-line and contained in workspaces, the job of maintaining it was turned over to the Registrar. Two terminals were installed in his office and MIS personnel trained two of his staff to use the terminals to update, add and delete student records. The system allowed us to produce mailing labels, a student directory and transcripts directly on line at the terminals in the Registrar's Office.

This system worked well for a small number of students and a limited amount of reporting but it soon became necessary to expand the student records system. A large file handling facility was added to the APL system in early 1972 and the student system was expanded to take advantage of the new capabilities. A student file was created outside the workspace and generalized retrieval and updating routines were written in APL. The file design allowed for 150 data elements which were completely variable in length. The user was prepared to enter about 100 elements so we had room for growth. On-line random access to any record in the file was possible through an index system very similar to IBM's ISAM indexes. The file could also be accessed in a sequential manner from a batch program.

The file updating routines were designed to be completely table-driven. A list of data elements desired to be updated is provided to the system and given a name. For example, the student schedule request form contains a total of about ten different elements which need to be updated during the registration process. The element numbers are entered into the system and given a name of Form 3. The terminal operator then calls out the generalized updating routine and specifies that she is using Form 3. The program then prompts the user for the student ID of the record to be updated, checks to make sure it is on the file and proceeds to prompt the user for each of the data elements on the Form 3 list by name. As each piece of information is entered, it is checked for validity according to values which are stored in an edit table. After finishing with one student, the file is updated and another student ID is requested. Any number of student records can be updated using this form and then another form can be specified and updating continued. Up to 99 different forms can be entered into the system. In general, each form corresponds to a source document which passes through the Registrar's office. Also, a new element can be defined to the system simply by adding an entry to the table containing the data element numbers and names and by providing its editing requirements to the edit table.

As in the original system, the file updating procedure is handled completely by the Registrar through the terminals located in his office.

The transcript process used in the first system was modified to access the new on-line file and a fee assessment procedure was implemented. These two applications were originally implemented as completely on-line procedures. It soon became apparent that the volume of students, which had now increased to 1200, could not be handled by these procedures because of the slowness of the terminal output capability. For this reason, a remote batch entry system was developed through the APL file system.

It was now possible to produce transcripts and fee assessments on-line and put the results on a file. A batch program could then be submitted to have the file printed on the high-speed printer at the CCC. It was then necessary to transport the printed output back to GSU from the CCC. For this purpose, we have employed a commercial courier service.

Since the student file is accessible from the batch as well as on-line, it is possible to write background programs which produced reports, print them at CCC and have them delivered back to GSU. We are now producing a total of ten or twelve different reports on a scheduled basis from the master file, including mailing labels, class lists, advisor lists and many more. We have also rewritten the fee assessment program to produce student billings completely from the background.

This is where the student records system stands today. The file is being updated and transcripts are being produced on-line. All reports, including fee assessments, are being produced in an offline mode and the transcripts are also being printed offline. We are planning some more improvements to the system and are in the process of implementing some of them. One of our main concerns is the delivery of output to the GSU campus. Funds have finally become available for the rental of an RJE terminal and we are planning to install it sometime in January, 1974.* This will give us the ability to print all of our batch output at GSU and eliminate the necessity for a courier service.

We are also redesigning the student file so that the master file will be stored on magnetic tape and be updated periodically by means of transactions. A subset of the master file will remain on line for terminal retrieval of individual student information. All reports will be produced offline from the master file, including fee assessments and transcripts.

In order to implement the transcript process using this new student file, we are also creating a learning module file which will have an indexed sequential organization and be accessible in a direct mode both from batch and from APL. Updating of this file will be handled through transactions from the batch, however, just as for the student file. We expect that the new system will relieve some of the heavy

*Since this paper was presented, the RJE station has been installed and is now working.

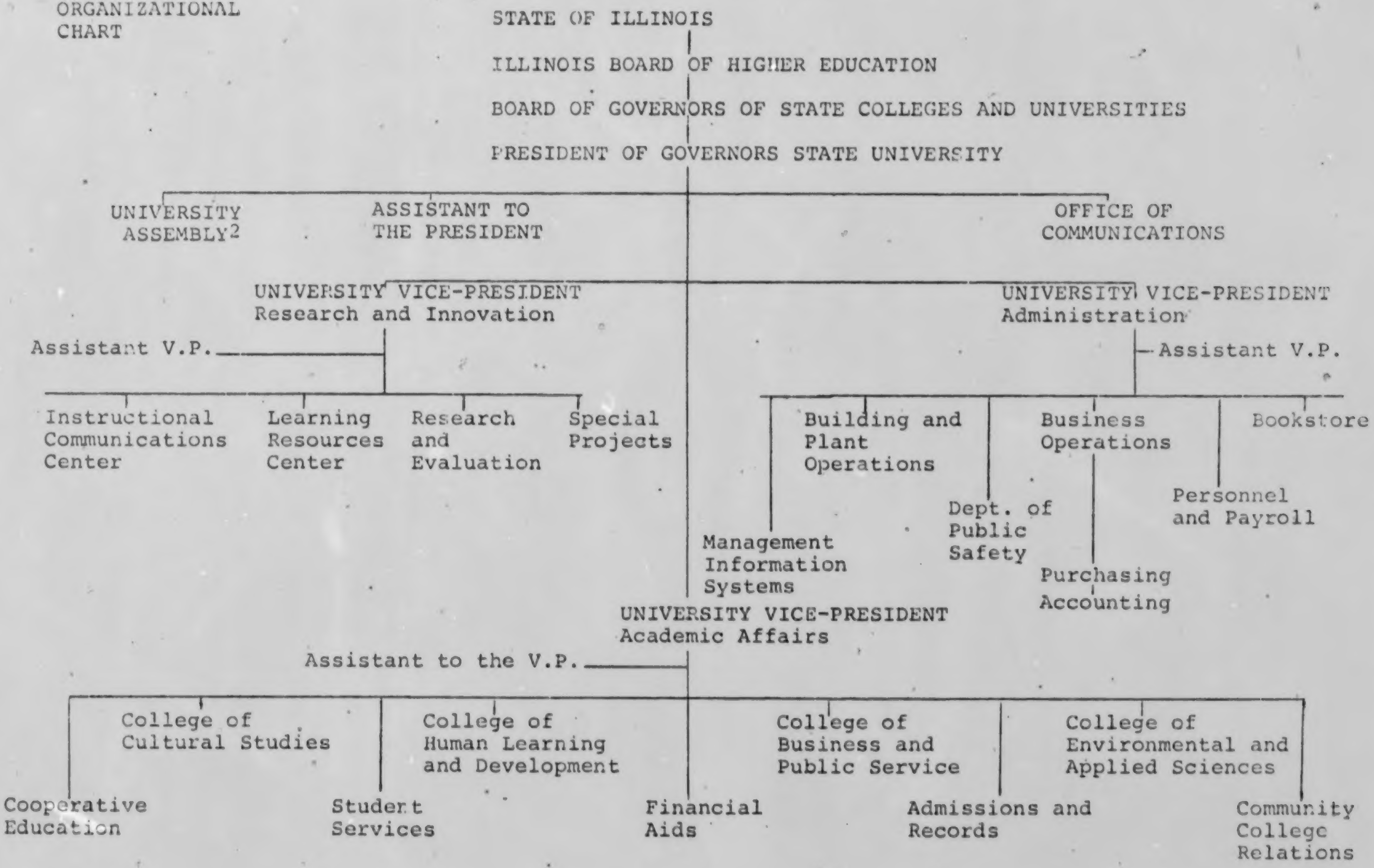
administrative load on the APL system and improve the response time for the instructional users. We intend to continue the practice of letting the Registrar's office enter all the data for maintenance of the file. We also plan to automate the procedure for requesting specific reports so that they can maintain closer control over which reports are run when. We are also planning for a mini-computer to be installed on the GSU campus. We expect this to do two things for us in relation to the student records system: to act as a store and forward type device by handling all the prompting and data entry locally, store the information temporarily, and then forward it to the central computer through the RJE mode for updating of the master files, and to provide a local data retrieval capability by maintaining an abbreviated version of the master file at GSU which would be refreshed daily or as often as the master file is updated.

CONCLUSIONS

In summary, we feel that our use of APL has been very beneficial to the university in the initial establishment of computing services to both the instructional and the administrative users. We have operated with a drastically reduced budget and a computer shared with two other institutions and have still been able to supply a time-sharing service to our academic users and to provide user-controlled file updating and information retrieval to administrative users. We have furnished our users with the opportunity to become involved with the computing procedure. This mode of operation has also forced the staff of the computing center, when designing and implementing a system, to consider the total system including the people involved. We feel that our use of APL has forced us to be user oriented in our design of administrative systems. This can mean only good things for the computing effort at Governors State University.

UNOFFICIAL
ORGANIZATIONAL
CHART

ORGANIZATION OF GOVERNORS STATE UNIVERSITY[1]


STATE OF ILLINOIS
ILLINOIS BOARD OF HIGHER EDUCATION
BOARD OF GOVERNORS OF STATE COLLEGES AND UNIVERSITIES
PRESIDENT OF GOVERNORS STATE UNIVERSITY
UNIVERSITY ASSEMBLY2
ASSISTANT TO THE PRESIDENT
OFFICE OF COMMUNICATIONS
UNIVERSITY VICE-PRESIDENT Research and Innovation
Assistant V.P.
Instructional Communications Center
Learning Resources Center
Research and Evaluation
Special Projects
UNIVERSITY VICE-PRESIDENT Administration
Assistant V.P.
Management Information Systems
Building and Plant Operations
Dept. of Public Safety
Business Operations
Purchasing
Accounting
Personnel and Payroll
Bookstore
UNIVERSITY VICE-PRESIDENT Academic Affairs
Assistant to the V.P.
Cooperative Education
College of Cultural Studies
Student Services
College of Human Learning and Development
Financial Aids
College of Business and Public Service
Admissions and Records
College of Environmental and Applied Sciences
Community College Relations

---

1 Does not indicate various advisory councils or community councils serving each college.

2 University Assembly consisting of students, faculty, support, civil service and community personnel is the policy formulating and recommending body of the University.

Figure 1

THE DESIGN OF A GENERALIZED MINI-COMPUTER BASED COLLEGE INFORMATION SYSTEM

George A. Mapp
Director
Data Processing
CUNY/Medgar Evers College

A GENERALIZED MANAGEMENT INFORMATION SYSTEM FOR MINI-COMPUTERS

A COMPONENT OF A SEMI-AUTOMATIC FAIL-SAFE/FAIL-SOFT INFORMATION SYSTEM

George A Mapp

I. INTRODUCTION

When the PDP-11/45 configuration was ordered for Medgar Evers College it was decided that the administrative applications and the interactive-BASIC instructional applications would have to run concurrently to insure full utilization of the computer. The PDP-11 configuration described in Table I can adequately support on-line integrated data base systems which a few years ago were considered for implementation only on medium-size to large systems costing 10 or more times the price of the present configuration.

The fail-safe/fail-soft concept we are employing is to have a system requirement of continous operation regardless of hardware or software failures. The City University of New York recently installed a 2 million byte IBM 370/165.

The system will be used remotely via low-speed and RJE terminals by all or most of the colleges in the nineteen college City University System. The system is operating under ASP version 3 and has been running the text-editing system, WYLBUR. At some later date the system will be replaced by and IBM 370/168.

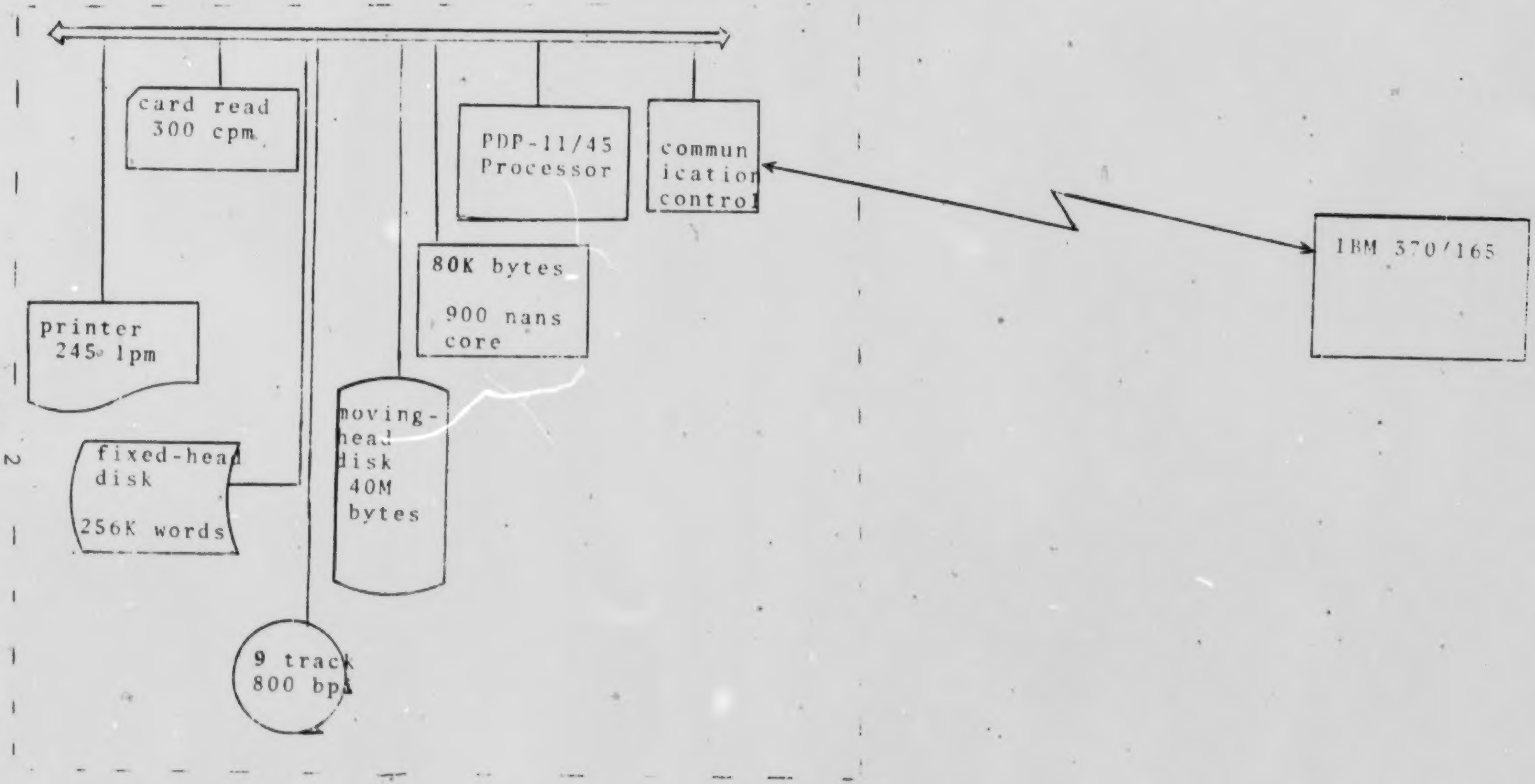
The day-to-day program accesses and file transactions will be interactively processed on the PDP-11. At the end of the day, the transaction log file will be sent to the 370 over telephone lines while operating the PDP-11 in the RJE mode. A device-driver for the PDP 11 will be modified to allow the tape drive to simulate the card reader. Thus, the log tape will be rewound and transmitted to the 370.

If there is a PDP-11 failure, the program accesses the file transactions are to be entered on the 370 via low speed terminals. Using the WYLBUR system the cataloged programs can be called and transaction data can be entered on-line into a temporary transaction file. This temporary file is later updated in batch mode. The PDP-11 transaction log tape will contain the transactions up to the point of system failure. If there is system failure with the 370 the PDP-11 will continue with its normal processing.

The system development stages are summarized as follows: a) document and information analysis; b) data element selection; c) general mini-computer system design; d) new mini-computer program specifications, design conversion programs test modified WCIS programs and file structure, creation for mini-computer system; e) data base creation for mini-computer

TABLE I.

PDP-11/45 CONFIGURATION & RJE CONNECTION


card read
300 cpm
PDP-11/45
Processor
commun
ication
control
IBM 370/165
80K bytes
900 nans
core
printer
245 lpm
moving-
head
disk
40M
bytes
fixed-head
disk
256K words
9 track
800 bpi

system; f) implement new programs for mini-computer systems; g) implement operational mini-computer systems, write specifications for 370 linkage report generation and conversion programs, and modify 370 system data-definition table; h) test 370 program modifications; i) create redundant data base on 370; j) test 370 system and k) run with dual system.

The external specifications for the system called for a combined reporting and information system which integrated several college administrative applications. The applications were chosen from studying the information in the college. The general goal is a system that will bring all the information together and make it accessible to all functional groups within the College. The specifications call for a system that is both on-line and batch with generalized maintenance, retrieval, sort and report programs.

Because of our small staff size, it would be impossible to run a traditional data processing department. Thus, we have a system requirement that practically all data input and report preparation can be performed by the user departments. The goal is to decrease redundancy of work among departments.

Thus, the computer center will be responsible for the integrity of the hardware and software system and the users will be responsible for the maintenance of the data. The project began approximately a month ago and we are currently at stage d. Steps a through c were performed by the writer and step d is currently being performed by the writer and a part-time student. Due to inadequate staffing we will "violate the rules" of employing a study and design team approach. However, we expect that our cost-effectiveness will be at least four times the industry standard. Several projects within the IBM Corporation have demonstrated drastic reductions in systems development manpower cost by abandoning the traditional systems development organizational structure.

By employing generalized information systems concepts we plan to demonstrate that with less than two FTE persons dedicted to the task, the mini-computer version of the information system will be fully implemented in less than one year. Within 1½ years we hope to have implmented a semi-automatic fail-safe/fail-soft system. Both systems should be relatively comparable to existing higher education management systems, which in some cases had an implementation cost of $500,000 or more in man-hours and/or software consulting services.

II. DOCUMENT AND INFORMATION ANALYSIS

Unlike simple data processing systems which are transaction oriented such as a payroll check printing program or integrated file-oriented systems such as a corporate personnel data system which is geared toward processing multi-input and multi-output data within a single class of application; an information system usually processes data concurrently for a whole range of applications. In this case the system is to process relevant data relating to all the required administrative institutional research applications which utilize the data in the files.

The first step was developing a graph model of the inter-department flow of information throughout the college. Figure 1 is a gross simplication of this model. This model was kept in mind during the analysis of required reports.

Medgar Evers College being a public institution and part of a nineteen college City University System has a vast amount of reports to produce per year. Besides the traditional internal reports for the admissions office, registrar's office, business office and administration; there are also required reports for the Central City University Office of Data Collection, New York City Controller's Office, U. S. Office of Education, New York City Board of Higher Education and New York State.

The document and information analysis procedure consisted of collecting copies of all the internal reports generated by the college. The next step was to list all the data elements required to produce the reports, their frequency of occurence, their sorting order and the usual length of the data elements. Figure 2 lists one sheet from document analysis procedure.

III. DATA ELEMENT SELECTION

The choice of data elements is primarily from the information analysis procedure above. In addition, the National Center for Higher Education Management Systems (NCHEMS) has produced a series of documents, recommendations, standards, suggested data elements for administrative and institutional research files and analytical procedures for management analytical procedures for management systems within colleges and universities. Many additional data elements were obtained from NCHEMS suggestions.

Table II shows the record layout of the student data base (CSB). One restriction of the PDP-11 implementation of BASIC plus operating under RSTS is the fixed disk record size which must be some power of two. Thus, there will be a need for several auxillary student files to get around the need for variable length records for such files as course history or advanced standing, which tend to vary in length as a function of the number of courses accumulated.

Since these two files in particular are generally used only during transcript printing they can be stored on tape. Tables III, IV, and V show the current registration file (CL I), course history file (CHF), course title file (TIT) and advanced standing fiel (ADV) respectively. Figure 3 is a copy of a typical student transcript that is produced on our current student records system and is run at Staten Island Community College of the City University.

This IBM 360/30 based system is an adaptation of a system originally designed by Tom Braniff of that College. The advanced standing text appears between the curriculum title and course information on a full length transcript.

Figure 1

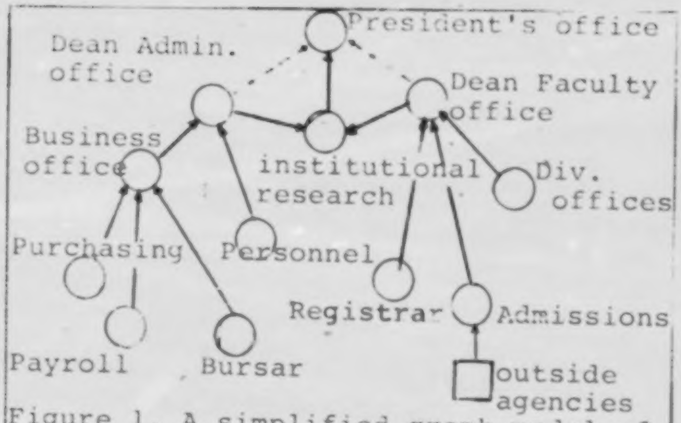

President's office
Dean Admin. office
Dean Faculty office
Business office
institutional research
Div. offices
Purchasing
Personnel
Registrar
Admissions
Payroll
Bursar
outside agencies

Figure 1. A simplified graph model of college's management information flow.

REPORT NAME yearly
data element virtual element freq.

26. HEDS: Residence & Migration 2
special programs code
special category code
total credits

27. Advisor's List 3
advisor's code
advisor's name
student curriculum
student name
grade point avg.
student classification

28. HEGIS: Credit by Subject Area 2
curriculum code
HEGIS program structure code
sex code
classification code

Figure 2. An exerpt from a sheet of the document analysis procedure.

TABLE II: STUDENT DATA-BASE

Element Number	Element	length	position	type	mnemonic	print editing	date edit
1	ID Code	6	1 - 6	char	ID		AN
2	name	23	7 -29	char	NM		ANB
3	social security	9	30 -38	char	SS		N
4	Bursar serial number	6	39 -44	char	BU		N
5	street & number	22	45 -66	char	ST		AN
6	city and state	19	67 -85	char	AZ		A
7	zip code	5	86 -90	char	ZI		N
8	sex code	1	91	char	SX		A
9	birth date D-M-Y	6	92 -97	char	BR	99-99-99	N
10	college admit date	3	98-100	char	CA		N
11	high school code	6	101-106	char	HS		N
12	high school grad. date	3	107-109	char	GR	X-XX	N
13	1st college choice cd	2	110-111	char	CH		N
14	1st curriculum choice	2	112-113	char	FI		N
15	curriculum code	2	114-115	char	CC		N
16	health alert code	1	116	char	HL		AN
17	local phone number	7	117-123	char	PH	99-99-99	AN
18	dependency code	1	124	char	DC		AN
19	welfare code	1	125	char	WC		AN
20	special category code	1	126	char	SC		A
21	special programs code	1	127	char	SP		A
22	financial aid code	1	128	char	FA		A
23	advisor's code	3	129-131	char	AC		N
24	college start date	3	132-134	char	SD	X-XX	N
25	source from code	1	135	char	SF		AN
26	college leave date	3	136-138	char	LD	X-XX	N
27	leave/reason code	1	139	char	LR		A
28	ethnicity code	1	140	char	EC		AN
	high school credits						
29	English	2	141-142	int.	EN		N
30	social science	2	143-144	int.	SL		N
31	1st language	2	145-146	int.	L1		N
32	2nd language	2	147-148	int.	L2		N
33	science	2	149-150	int.	SN		N
34	number in family	2	151-152	int.	NF		N
35	admissions average	2	153-154	int.	AA	99.99	N
36	grade-point average	2	155-156	int.	GP	9.99	N
37	classification code	1	158	char	CL		N
38	high-schl math avg.	2	159-160	int.	MA	99.99	N
39	English average	2	161-162	int.	EA	99.99	N
40	credits completed	2	163-164	int.	CT	99.99	N
41	major subj. credits	2	165-166	int.	MJ	99.99	N
42	liberal arts credits	2	167-168	int.	LA	99.99	N
43	current sem. credits	2	169-170	int.	SC	99.99	N
44	current sem. hours	2	171-172	int.	HR		N
45-55	course hash Nos	(10 × 2)	173-192	int.	C0-C9	99.99	N
56	advance standing date	3	193-195	char	AD	X-XX	N
57	adv stand end date	3	196-198	char	DE	X-XX	N
58	old course start date	3	199-201	char	CD	X-XX	N
59	old course end date	3	202-204	char	CE	X-XX	N
60	session code	1	205	char	SE		A
61	fee code	1	206	char	FC		A

element number	element	length	position	type	mnemonic	Print Editing	Data edit
62	tuition fee	4	207-210	flt.	TF	$ 999.99	N
63	other fee	4	211-214	flt	OF	999.99	N
64	total fee amount	4	215-218	flt	TL	9999.99	N
65	family income	4	219-222	flt	IN	$99,999.99	N
66	Part-full time code	1	223	char	FP		
		33	224-256				

TABLE III: CURRENT REGISTRATION FILE

	Element	Length	Position	Type	Mnemonic	Print Editing	Data Edit
1	ID Code	6	1-6	C	ID		AN
2	sem/year date:MYY	3	7-9	C	SY	X-XX	N
3	division code	1	10	C	DT		N
4	course: key-No.-section	9	11-19	C	CS	X(7)-XX	ANB
5	course hash number	2	20-21	I	C	XXXX	N
6	credits	2	22-23	I	CR	99.99	N
7	hours	2	24-25	I	H1	99.99	N
8	grade	2	26-27	C	GD		AS B
9	part/full time code	1	28	C	FP		A
10	sex code	1	29	C	SX		A
11	classification code	1	31	C	CL		N
12	curriculum code	2	32-33	C	CC		N
13	special programs code	1	34	C	SP		A
14	special category code	1	35	C	SC		A
15	level code	1	36	C	LE		N
16	session code	1	37	C	SE		A
17	name	23	38-60	C	NM		AB
		4	61-64				

TABLE IV: COURSE HISTORY FILE

Element Number	Element	Length	Position	Type	Mnemonic	Print Editing	Data Edit
1	ID code	6	1-6	C	ID		AN
2	semester/year date MYY	3	7-9	C	SY	X-XX	N
3	course: key-number	7	10-16	C	CS	X(7)	AN
4	credits	2	17-18	I	CR	99.99	N
5	grade	2	19-20	I	GD		ASB
6	level code	1	21	C	LE		N
7	type of course code	1	22	C	TY		A
8	course title hash No.	2	23-24	I	C	XXXX	N
9	division code	1	25	C	DT		N
			26-32				

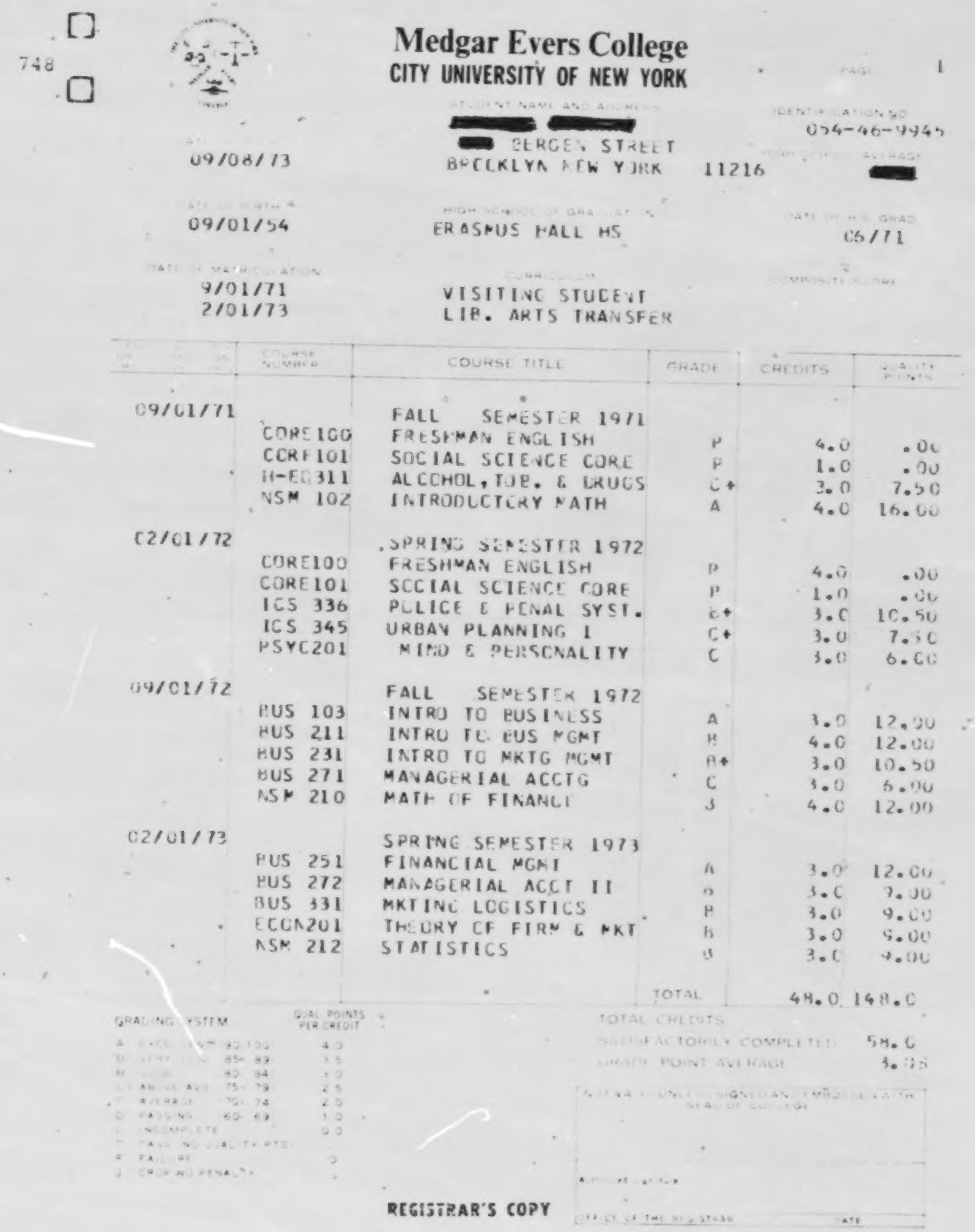
TABLE V: COURSE TITLE AND ADVANCED STANDING OR COMMENT TEXT FILES

Element Number	Element	Length	Position	Type	Mnemonic	Print Editing	Data Edit
1	ID code	6	1-6	C	ID		AN
2	Course	7	7-13	C	CS	X(7)	AN
3	2nd Course	7	14-20	C	C2	X(7)	AN
4	Course title	22	21-42	C	TI		AN
5	Continue code	1	43	C	CN		A
6	Title expand	21	44-64	C	T		AN

Course title file

1	ID code	6	1-6	C	ID		AN
2	sem/year date MYY	3	7-9	C	SY	X-XX	N
3	count	1	11	C	TT		AN
4	adv stand or text 1	20	12-31	C	A1		ANB
5	adv stand/text 2	20	32-51	C	A2		ANB
6	adv stand/text 3	20	52-71	C	A3		ANB
7	adv stand/text 4	20	72-91	C	A4		ANB
8	adv stand/text 5	20	92-101	C	A5		ANB
9	adv stand/text 6	20	102-121	C	A6		ANB
10	type code	1	122	C	CM		ANB
			123-128				

Advanced standing/comment file



748

Medgar Evers College
CITY UNIVERSITY OF NEW YORK

PAGE 1

STUDENT NAME AND ADDRESS
[redacted] [redacted]
[redacted] BERGEN STREET
BROOKLYN NEW YORK 11216

IDENTIFICATION NO
054-46-9945

HIGH SCHOOL AVERAGE
[redacted]

09/08/73

DATE OF BIRTH
09/01/54

HIGH SCHOOL OF GRADUATION
ERASMUS HALL HS

DATE OF H.S. GRAD
06/71

DATE OF MATRICULATION
9/01/71
2/01/73

CURRICULUM
VISITING STUDENT
LIB. ARTS TRANSFER

COMPOSITE SCORE

[illegible]	COURSE NUMBER	COURSE TITLE	GRADE	CREDITS	QUALITY POINTS
09/01/71		FALL SEMESTER 1971			
	CORE100	FRESHMAN ENGLISH	P	4.0	.00
	CORE101	SOCIAL SCIENCE CORE	P	1.0	.00
	H-ED311	ALCOHOL,TOB. & DRUGS	C+	3.0	7.50
	NSM 102	INTRODUCTORY MATH	A	4.0	16.00
02/01/72		SPRING SEMESTER 1972			
	CORE100	FRESHMAN ENGLISH	P	4.0	.00
	CORE101	SOCIAL SCIENCE CORE	P	1.0	.00
	ICS 336	POLICE & PENAL SYST.	B+	3.0	10.50
	ICS 345	URBAN PLANNING I	C+	3.0	7.50
	PSYC201	MIND & PERSONALITY	C	3.0	6.00
09/01/72		FALL SEMESTER 1972			
	BUS 103	INTRO TO BUSINESS	A	3.0	12.00
	BUS 211	INTRO TO BUS MGMT	B	4.0	12.00
	BUS 231	INTRO TO MKTG MGMT	B+	3.0	10.50
	BUS 271	MANAGERIAL ACCTG	C	3.0	6.00
	NSM 210	MATH OF FINANCE	B	4.0	12.00
02/01/73		SPRING SEMESTER 1973			
	BUS 251	FINANCIAL MGMT	A	3.0	12.00
	BUS 272	MANAGERIAL ACCT II	B	3.0	9.00
	BUS 331	MKTING LOGISTICS	B	3.0	9.00
	ECON201	THEORY OF FIRM & MKT	B	3.0	9.00
	NSM 212	STATISTICS	B	3.0	9.00
			TOTAL	48.0	148.0

GRADING SYSTEM		QUAL POINTS PER CREDIT
A EXCELLENT	90-100	4.0
B+ VERY GOOD	85-89	3.5
B GOOD	80-84	3.0
C+ ABOVE AVG	75-79	2.5
C AVERAGE	70-74	2.0
D PASSING	60-69	1.0
I INCOMPLETE		0.0
P PASS NO QUALITY PTS		
F FAILURE		0
J DROP NO PENALTY		

TOTAL CREDITS SATISFACTORILY COMPLETED 58.0
GRADE POINT AVERAGE 3.08

NOT VALID UNLESS SIGNED AND EMBOSSED WITH SEAL OF COLLEGE

AUTHORIZED SIGNATURE

OFFICE OF THE REGISTRAR DATE

REGISTRAR'S COPY

Figure 3. A typical student transcript with advanced standing component left out and reduced in length for this paper.

TABLE VI: COURSE FILE

	Element	Length	Position	Type	Mnemonic
1	ID Code	6	1 - 6	C	ID
2	Division code	1	7	C	DT
3	Course level	1	8	C	LE
4	Course type code	1	9	C	TY
5	Hash number	2	10 - 11	I	C
6	Course key, no., section	9	12 - 20	C	CS
7	CUNY Dept. code	3	21 - 23	C	CU
8	Course title	25	24 - 48	C	TI
9	Credits	2	49 - 50	I	CR
10	Hours	2	51 - 52	I	H1
11	Suggested class limit	2	53 - 54	I	LM
12	Number in class	2	55 - 56	I	NC
13	1st instructor code	3	57 - 59	C	IC
14	2nd instructor code	3	60 - 62	C	I2
15	1st instructor PT/FT	1	63	C	PF
16	2nd instructor PT/FT	1	64	C	PT
17	Instructor name	23	65 - 88	C	IN
18	Day code	2	89 - 90	C	DY
19	Starting time	2	91 - 92	I	ST
20	Ending time	2	93 - 94	I	ET
21	Building code	2	95 - 96	C	BC
22	Room number	4	97 -100	C	RN
23	Number in class last sem	2	101-102	I	NC
24	Number/2-sem-ago	2	103-104	I	N2
25	Number sections last sem	2	105-106	I	NS
26	Number sections 2 sem ago	2	107-109	I	S2
27	Semester/year date	3	110-112	C	SY
28	Joint course number	7	113-119	C	JC
29	HECIS program structure	8	120-127	C	HG
30	Continue code	1	128	C	CN

TABLE VII PERSONNEL DATA-BASE

750

element	Length	position	type
1 ID code	6	1 - 6	char
2 name(Last, first)	22	7 -28	char
3 middle initial	1	29	char
4 social security	9	30 -38	char
5 CUNY title code	3	39 -41	char
6 CUNY dept. code	3	42 -44	char
7 street & number	22	45 -66	char
8 city and state	19	67 -85	char
9 zip code	5	86 -90	char
10 sex code	1	91	char
11 birth date DDMMYY	6	92 -97	char
12 full/part time code	1	98	char
13 highest degree code	3	99 -101	char
14 next highest degree	3	102-104	char
15 third highest degree	3	105-107	char
16 ethnic code	2	108-109	char
17 tenure status	1	110	char
18 tenure title	3	111-113	char
19 tenure date DDMMYY	6	114-119	char
20 civil service status	1	120	char
21 initial title code	3	121-123	char
22 5th title code	3	124-126	char
23 4th title code	3	127-129	char
24 3rd title code	3	130-132	char
25 2nd title code	3	133-135	char
26 current title code	3	136-138	char
27 initial appt. date	6	139-144	char
28 5th title date	6	145-150	char
29 4th title date	6	151-156	char
30 3rd title date	6	157-162	char
31 2nd title date	6	163-168	char
32 current title date	6	169-174	char
33 time basis current salary	1	175	char
34 time basis 2nd sal.	1	176	char
35 time basis initial title	1	177	char
36 current salary	4	178-181	flt.
37 previous salary	4	182-185	flt.
38 initial salary	4	186-189	flt.
39 % current salary	2	190-191	int.
40 % previous salary	2	192-193	int.
41 % initial salary	2	194-195	int.
42 multiple appt. code	1	196	char
43 personnel action code	1	197	char
44 division code	1	198	char
45 last job held code	1	199	char
46 highest degree code	3	200-202	char
47 2nd highest degree	3	203-205	char
48 3rd highest degree	3	206-208	char
49 1st highest institution	4	209-212	char
50 2nd institution code	4	213-216	char
51 3rd institution code	4	217-220	char
52 high school state code	2	221-222	char
53 grad date 1st degree: YY:	2	223-224	char
54 grad date 2nd degree	2	225-226	char
55 grad date 3rd degree	2	227-228	char

element	Length	position	Type
56 retirement system code	1	229	char
57 HEDS title code	1	230	char
58 HEDS ed. code	1	231	char
59 HEGIS title code	2	232-233	char
60 HEGIS salary code	2	234-235	char
61 Bldg. Code	2	236-237	char
62 Room number	4	238-241	char
63 office phone	7	242-248	char
64 office extension	3	249-251	char
65 leave date	2	252-253	int.
66 leave code	1	254	char
67 pension deduction	1	255	char
68 FLSA code	1	256	char

PERSONNEL DATA BASE

TABLE VIII: AUXILLIARY PERSONNEL FILE

element	length	position	type
1 ID code	6	1 - 6	char
2 time basis	1	7	char
3 current title	3	8 - 10	char
4 current salary	4	11 - 14	flt
5 tax class	1	15	char
6 current hourly rate	2	16 - 17	int
7 current acc. salary	4	18 - 21	flt
8 hours accumulated	2	22 - 23	int
9 pension / deduction	1	24	char
10 vacation days	2	25 - 26	int
11 comp. days	2	27 - 28	int
12 sick days	2	29 - 30	int
13 overtime rate	2	31 - 32	int
14 overtime hours	2	33 - 34	int
15 overtime accumulated	4	35 - 38	flt
16 late hours	2	39 - 40	int
17 teaching work load	2	41 - 42	int
18 research work load	2	43 - 44	int
19 college admin. wk ld	2	45 - 46	int
20 dept admin work load	2	47 - 48	int
21 other work load	2	49 - 50	int
22 other wk load code	1	51	int
23 current teaching hrs	2	52 - 53	int
24 current tchg credits	2	54 - 55	int
25 current student hrs	2	56 - 57	int
26 health code	1	58	char
27 HEGIS major subject	4	59 - 62	char
28 activity outcome cd	2	63 - 64	char

TABLE IX: FACILITIES FILE

element	length	position	type
1 ID code	6	1 - 6	char
2 space type code	2	7 - 8	char
3 building code	2	9 - 10	char
4 room number	4	11 - 14	char
5 classroom capacity	2	15 - 16	int
6 number of stations	2	17 - 18	int
7 number of hrs use	2	19 - 20	int
8 division assigned	1	21	char
9 square feet	2	22 - 23	int
10 hours not used	4	24 - 27	flt

TABLE X: INVENTORY FILE

	Element	Length	Position	Type	Mnemonic
1	ID code	6	1 - 6	C	ID
2	Date received DD-MM-YY	6	7 -12	C	DR
3	Purchase order No.	6	13-18	C	PO
4	Tag number	6	19-24	C	TG
5	Purchase price (net)	4	25-28	F	PP
6	Equipment code	2	29-30	C	TC
7	Bldg. code	2	31-32	C	BC
8	Room number	4	33-36	C	RN
9	Division code	1	37	C	DT
10	Vendor code (optional)	3	38-40	C	VC
11	Item description	5	41-55	C	IT
12	Vendor abbreviation	9	56-64	C	VA

TABLE XI: ACCOUNTS/PURCHASING FILE

element	length	position	type
1 ID code	6	1 - 6	char
2 activity number	4	7 - 10	char
3 expense code	3	11 - 13	char
4 fund class	2	14 - 15	char
5 project number	4	16 - 19	char
6 source of funds	3	20 - 22	char
7 division charged	1	23	char
8 program charged	6	24 - 29	char
9 functional class cd	2	30 - 31	char
10 fiscal year YY	2	32 - 33	char
11 net amount	4	34 - 37	flt
12 received code	1	38	char
13 equipment code	1	39	char
14 date ordered YYMMDD	2	40 - 41	int
15 date received	2	42 - 43	int
16 date paid	2	44 - 45	int
17 vendor code (opt)	3	46 - 48	char
18 vendor name	16	49 - 64	char

TABLE XII: DEPARTMENT FILE

element	length	position	type
1 ID code	6	1 - 6	char
2 CUNY dept code	3	7 - 9	char
3 program structure	6	10 - 15	char
4 division code	1	16	char
5 YTD salaries	4	17 - 20	flt
6 YTD FTE revenue	4	21 - 24	flt
7 YTD purchase tot	4	25 - 28	flt
8 YTD other OTPS exp	4	29 - 32	flt

TABLE XIII: BUDGET FILE

	element	length	pos.	type
1.	ID Code	6	1- 6	C
2.	session	2	7- 8	C
3.	line number	3	9-11	C
4.	title number	4	12-15	C
5.	project number	4	16-19	C
6.	source of funds	3	20-22	C
7.	amount	4	23-26	F
8.	CUNY title code -or	3	27-29	C
9.	expense code	3	30-32	C

A summary ADVSTAN course card appears in the old course file and is computed in the GPA as a "P" grade. The data elements in the 360/30 based system will be converted to the files CSB, CHF and ADV. All three files will be open for transcript printing on the PDP-11 based system. In addition, the table file TIT will also be open to print the course title on the transcripts. The course file (SUB) data elements appear in Table VI. The source of these elements include the registration reports, class schedules, faculty teaching load reports and NCHEMS suggested elements.

The data elements in the personnel data base (PRO) selected from the information analysis are primarily required for external reporting. All but two of the first 56 data elements of this file are required for the CUNY Office of Data Collection and Evaluation. The personnel data base is displayed on Table VII. Table VIII displays an auxillary personnel file. This second file was employed to avoid a personnel data base of 512 bytes (i.e., the next power of 2). This auxillary personnel file (STF) (MAN) contains benefits, summary payroll and NCHEMS recommended data elements. This file will be used primarily for the subset RRPM 1.6 programs, hourly and non-instructional staff time accounting, instructional staff workload computations and a subset faculty activity analysis. The facilities file (FCL) data element derived during the information analysis came primarily from course schedule and room utilization reports. Other elements were NCHEMS suggested and from CUNY requirements for square feet per staff member.

Table IX shows the facilities file. The purchasing/accounts file (PUR) data elements came from the purchase orders and New York City Controller's Office accounts. This file is primarily for accounts payable.

There will not be an accounts receivable file at this time. The City University Colleges are funded with tax-levied funds, based on FTE (full time equivalent) students. Partial tuition paying and tuition paying students. Partial tuition paying and tuition paying students are a small population. Thus, the fee income reports can easily be obtained from the student data base. In addition, the accounting of foundation grants and related aid is handled by the Central City University Research Foundation. Other receivables can be handled manually at this time. If there is a future need for an account receivable file it will be linked to the budget file and possibly to the department file. That is, the budget file will be debited and a department file would be debited, when appropriate, with non-tax levied monies.

The inventory file (INV) elements were selected by the writer in consultation with the fiscal office staff. Many of the elements will be transferred from the purchasing accounts file after a capital goods item is received and tagged. The inventory file appears in Table X. The elements of the budget file were selected by the writer in consultation with the fiscal office staff. Other elements selected in the information analysis came from only one report, namely the HEGIS Financial Statistics survey. Another minor influence on the budget file elements came from reviewing old special information request disseminating from the New York City Bureau of the Budget.

IV. GENERAL SYSTEMS DESIGN

Since the introduction of CAMPUS, a system developed by the University of Toronto, there have been several major integrated data base, planning and management information systems implemented. Many of these systems contain cost-simulation and projection models, but many do not provide processing of everyday administrative reports. Some later systems include: HELP/PLANTRAN, OASIS, TOTAL, SEARCH and HEPS. The number of man-years and cost outlays to implement these systems have often been substantial, and their current costs run from a few thousand to over $100,000.

The current system is an attempt to accomplish similar goals with a mini-computer system and a small staff. The system has seven major files, ie., student, staff, course, department, purchasing/accounts, budget, and facilities, as well as several other auxiliary files. Since no more than six files will be open concurrently for any one program, it can easily be handled by BASIC plus.

Figure 4 shows the general application flow within the system. File creation will consist of converting existing files to the system and creating new files. The facilities, budget, department, inventory and purchasing/accounts files will have to be created from scratch. The student, course title, advanced standing, and old course will be created by converting current tape records in EBCIDIC to ASCII. Files such as course and current registration will be created each semester.

Although the system contains several data bases, the file maintenance will not be as massive as it first appears. One reason is that many of the elements in some of the files are created virtually by computation or by other elements or the data is passed from files created by other agencies. For example, Figure 5 is a list of elements on the admissions tape received by the nineteen City University Colleges from the University Applications Processing Center, which is a central admissions operation. By comparing the elements on their list with the student data base, one can see that much of the new record creation is assisted by this tape of all applicants to the colleges.

The facilities file is created once, after which it is practically never updated. Much of the personnel data base is generated by elements collected for the University Office of Data Collection. The semester to semester changes in the course file usually include only instruction, building and room information. The student data base elements of the on-line created current registration file are generated "virtually" whenever the record being keyed matches a data base record.

The inventory file is created much the same way when a purchased item is received and tagged. The department file is essentially a summary file of information generated "virtually," based on calendar time, such as year-to-date purchase totals. Figure 6 shows a conceptual example of some of the linking between files in the system.

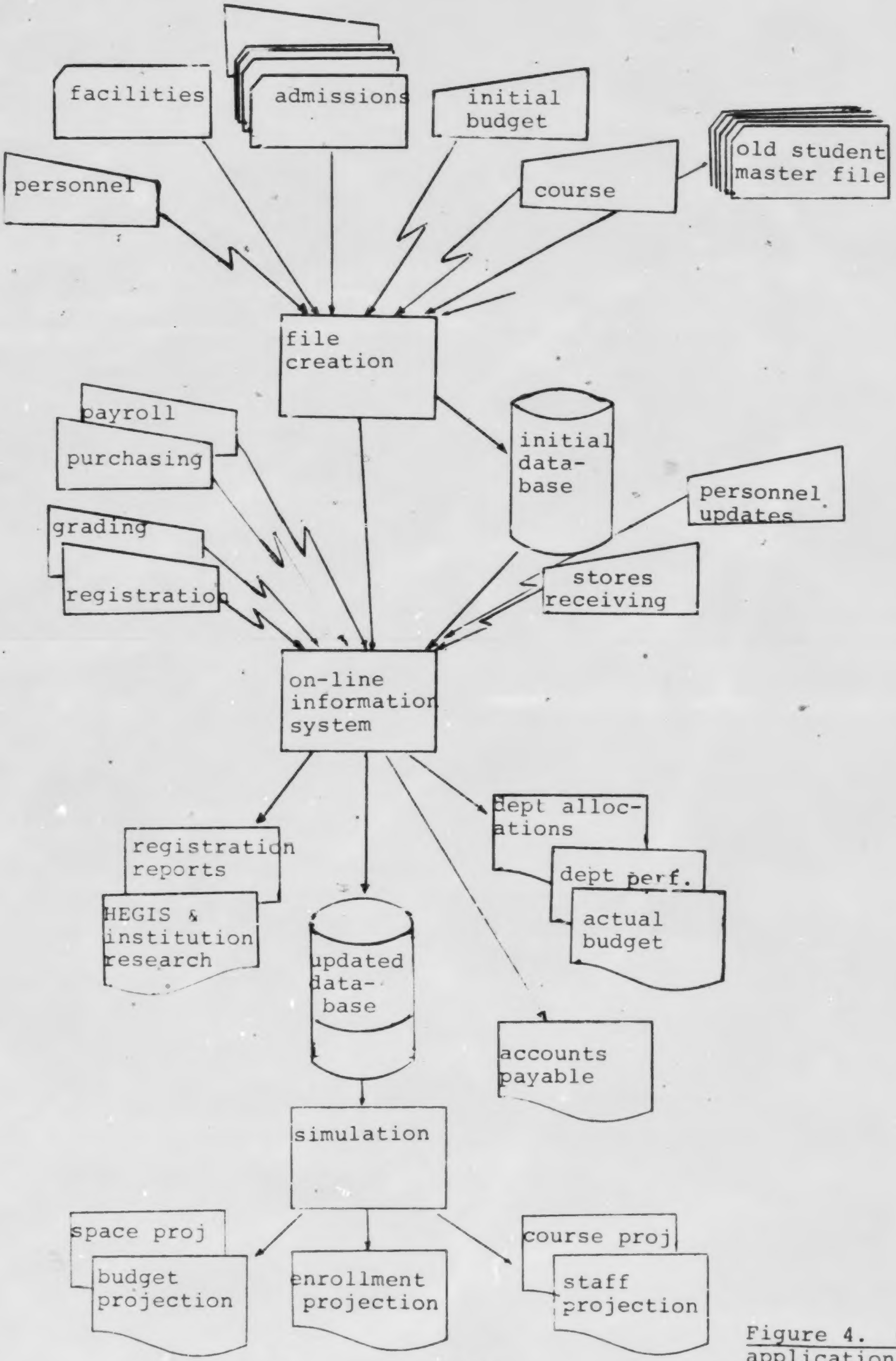

facilities
admissions
initial budget
personnel
course
old student master file
file creation
payroll
purchasing
initial data-base
personnel updates
grading
registration
stores receiving
on-line information system
registration reports
HEGIS & institution research
updated data-base
dept allocations
dept perf.
actual budget
accounts payable
simulation
space proj
budget projection
enrollment projection
course proj
staff projection

Figure 4. General application flow.

COLLAD D

COLLAD TAPE LAYOUT FEBRUARY 1974 ADMISSIONS

1-20	Name (Last, First)
21-30	High School
31	Sex (M,F)
32-34	CAA (xx.x)
35-37	Percentile (xx.x)
38-43	BLANK
44-49	Credits
44	E (English)
45	FL1 (1st Foreign Language)
46	FL2 (2nd Foreign Language)
47	M (Math)
48	Sc (Science)
49	SS (Social Studies)
50	BLANK
51-52	Total Academic Credits - 11 overpunch in 52 is .5
53	Math Symbol
54	Science Symbol
55	Health Alert
56	Residence Alert
57	Special Programs Eligibility
58	Allocation as S, C, B or X (below 70% HS avg., 1st ch. senior college)
59	BLANK
60-61	1st Choice College - 2 numbers
62-63	1st Choice Curric. - 2 numbers
64	1st Choice Session (1=Day; 2=Eve)
65-66	Allocation College - 2 numbers
67-68	Allocation Curric. - 2 numbers
69	Allocation Session (1=Day; 2=Eve)
70	Choice Number of Allocation (1-6)
71-90	Name (First, Last)
91-108	Address (Number and Street)
109-120	Address (City and State)
121-125	Zip
126-129	ETS
130-131	English Average
132-133	1st Foreign Language Average
134-135	Math Average
136-137	Science Average
138-139	Social Studies Average
140-170	BLANK
171-179	Social Security Number
180-189	BLANK
190-195	Birthdate (MMDDYY)
196-199	Date of Graduation (MMYY)
200	C
201	Veteran Status (V)
202-256	UNUSED

Figure 5. Data elements on the central admissions tape.

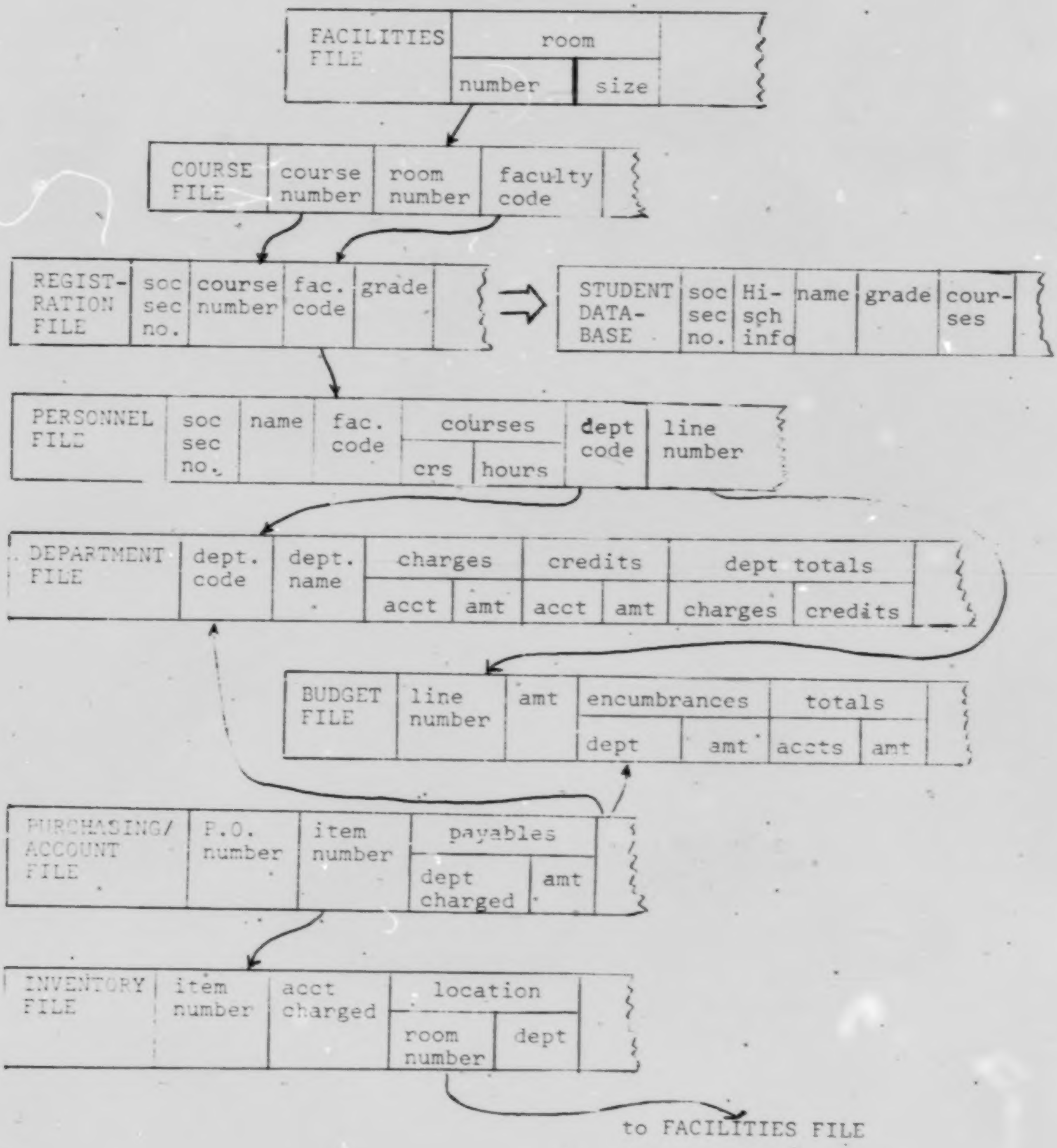

FACILITIES FILE
room
number
size
COURSE FILE
course number
room number
faculty code
REGISTRATION FILE
soc sec no.
course number
fac. code
grade
STUDENT DATA-BASE
soc sec no.
Hi-sch info
name
grade
courses
PERSONNEL FILE
soc sec no.
name
fac. code
courses
crs
hours
dept code
line number
DEPARTMENT FILE
dept. code
dept. name
charges
acct
amt
credits
acct
amt
dept totals
charges
credits
BUDGET FILE
line number
amt
encumbrances
dept
amt
totals
accts
amt
PURCHASING/ ACCOUNT FILE
P.O. number
item number
payables
dept charged
amt
INVENTORY FILE
item number
acct charged
location
room number
dept
to FACILITIES FILE

Figure 6. The relationship between administrative information system files.

The projection models will be of two levels, a simplified projection based on enrollment and projections from a subset version of the NCHEMS RRPM 1.6 model. The logic flow of RRPM 1.6 includes several steps such as developing an induced course load matrix, computing FTE faculty salaires, and calculating discipline and program cost. Another procedure borrowed from NCHEMS studies is the faculty activity analysis of work load data. All the programs for the RRPM and FAA are being developed locally.

The system will utilize the generalized DEC sort package which includes the programs SORT, MERGE and FILMAN. In addition, BASIC statistical and plotting programs are being modified to become more generalized. They will be added to the computation and print modules. These programs include: CORREL, MULREG, HIST, GRAPH, TTEST, ANOVA, CRFIT, PLOT2, RANTK, MEAN.SD and CHISQ. The purpose of these and similar program modules is improved data analysis and data presentation without having to write a complex program.

Figure 7 is a flow of the general systems logic. The logic is modular in design. Some programs may access all eight modules. The modules are: mainline program, optional pre-sort, edit routine, file maintenance, information retrieval, computation, optional post-sort, and report generation.

The several core programs from the Wabash College system play an important part in and are at the heart of the system. The Wabash program EDITA creates an attribute file, which is essentially a dictionary or data definition table, called ATRIB. The file ATRIB describes the file attributes: file name, type of information, number of data elements, number of records per block, file title, two byte mnemonic describing data element, starting location of element, and element length.

A standard subroutine in the Wabash programs which accesses the ATRIB file occupies lines 4000-9999 of a program. The integers N% and N1% contain records per block and record length, respectively. Mnemonics and filed descriptions appear in the arrays V2$ and V2%. Thus, the file definition subroutine looks up at the attributes of the file to be processed by a given program module. This approach allows for the creation of new files by simply interactively describing the structure to the program EDITA.

However, there are some additional attributes which I plan to add to the file, particularly to enhance generalized report generation and data editing. One of these is the data element number. This number can be used as an identification code for printing. Another enhancement is an array of the number of print positions per data element. A third will be an array of element column headings for printing, which will contain the text or code "1" indicating the use of mnemonic for the heading. A fourth modification is an array of PRINT-USING editing formats for printing purposes. Other modifications include an array of data element edit codes (A, alpha; B, blank; N, numeric and S, special character) in radis 50 character set. PDP-11 BASIC plus can recognize any combination of three of the four codes.

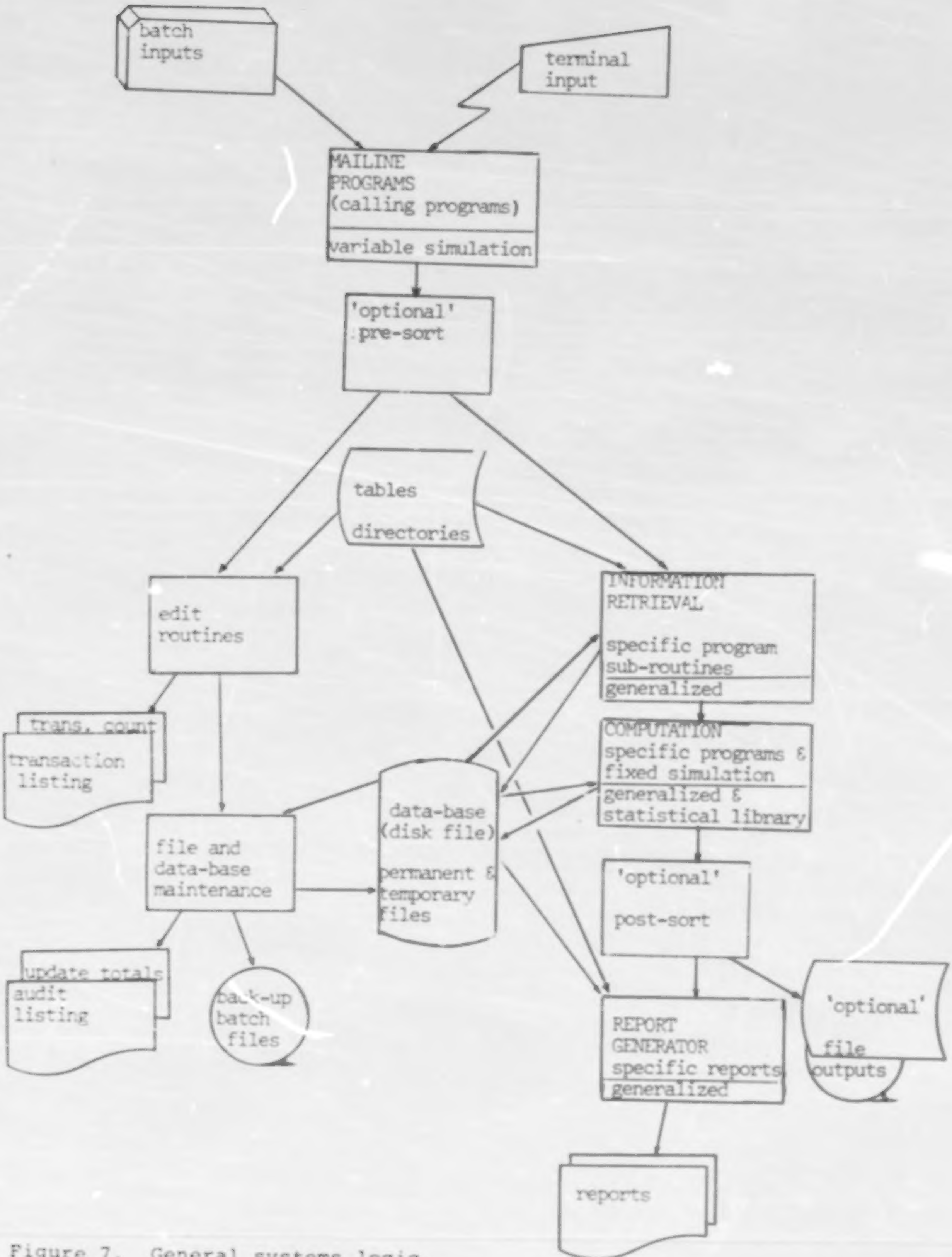

batch inputs
terminal input
MAILINE PROGRAMS (calling programs)
variable simulation
'optional' pre-sort
tables
directories
edit routines
INFORMATION RETRIEVAL
specific program sub-routines
generalized
trans. count
transaction listing
COMPUTATION specific programs & fixed simulation
generalized & statistical library
data-base (disk file)
permanent & temporary files
file and data-base maintenance
'optional'
post-sort
update totals
audit listing
back-up batch files
REPORT GENERATOR specific reports
generalized
'optional'
file outputs
reports

Figure 7. General systems logic.

For example, Figure 8 demonstrates the combination of ANB. A fifth modification is a list of source data element numbers or those "virtual" data elements which are created in a one-to-one transfer from one file to another. This array will contain file type mnemonic and element number within that file. Another form of "virtual" data element, namely, sliding elements, are also handled the same way. An example of a sliding element is when a person's job title changes, the old "current" title code is "moved" to the "previous" title code. Other more complex "virtual" elements and links will be handled with file maintenance subroutines. A final modification is an integer array of counters that count how often each element is used during the printing phase. In other words, at the end of the year, there will be a frequency count of element use. What follows is a brief description of the eight modules displayed in Figure 7.

Mainline Program

The mainline program will contain the necessary interactive dialog for all of or a particular program. The mainline program can call any number of the other routines on an as needed basis. The programs or routines within the other modules will be automatically called when there is a pre-determined sequence of steps for some commonly used applications. If the user has to decide between a series of alternatives which will be offered via an INPUT statement he will enter a code. This code will normally select which of the other modules will be accessed during processing. For example, the dialog for a class list program would have the user access the pre-sort to sort the current registration file in student within course and section order; the retrieval module will slect the data elements for printing; the totals subroutines from the computation module will be called for class and overall totals; the post sort is skipped and the report generator will interpret a string of data elements and spaces, access the ATRIB file and print the report.

Some specific programs such as user-written simulation equations or reports using derived data elements may not follow the normal sequence.

All data input is directed to the optional presort or directly to the edit routine. This input data will be associated with a code indicating the action to be taken; the action codes are: '1' add a record, '2' drop a record and '3' update or change an element in the record. All file accesses for display or printing are optionally directed to the presort or directly to the retrieval module.

Optional Pre-sort

The optional presort module will be called for data inputed in large volumes or batch data which is to be put in the sequence of the file to be updated. The sort programs are the DEC RSTS-11 Sort Programs. A required front-end routine in the mainline program will allow the sort keys to be optionally selected. An alternative to the DEC sort package is the Wabash SORTX program.

char		radix-50
A		003100
N		053600
B		006200
AN$\overline{B}$	=	062900

Figure 8. An example of radix-50 character set for three of the edit codes.

760
Figure 9, below, is a schematic of the revised attribute file structure.

FILE DESCRIPTION

file-name	file-type mnemonic	number of elements	number of records/block	file title	cluster size (blocking factor)

DATA-ELEMENTS' DESCRIPTORS

mnemonic	element number	**starting** location	element length	number of print positions
print headings	print editing formats	data-edit codes	source elements for "virtual" elements	element use frequency count

Figure 9. A schematic of the revised attribute file structure.

Edit Routines

This module accepts either the data directly from the mainline or after it has been presorted. This module is essentially an edit and pass procedure which also produced a transaction listing which counts and flags the data elements with data errors. The erroneous data does not pass. The flag also tells what action the error should take. This module is essentially the Wabash program's EDITX (sequential editing) and EDITR (random input).

The program modifications include: enabling the program to validate the data element against the edit codes in the modified attribute file; checking for missing elements in all code '1' inputs (i.e., adding a new record); indicating an erroneous ID code and indicating if the data is too large for the fixed element (value) it specifies. The final output is a transaction file with codes 1, 2 or 3 indicating what action is to be taken on the data within a given major key (ID code).

File Maintenance

The data which is inputted randomly passes through a routine called RANSEQ to put the file in ID code order. Next, a Wabash routine called FIX merges multiple transaction files into a single file and removes records with duplicate ID codes. A planned modification is to have the program also check action codes.

The file will be updated with a modified version of the Wabash UPDATX program. One modification is to create virtual data elements on multiple output files. For example, while registration is taking place, you can also virtually update the course file with class totals. This procedure will use the "virtual element source numbers" in the modified ATRIB file.

Output will be an exception audit listing of those data elements which pass each transaction phase but do not pass the file maintenance phase. These data elements or records will have an error list which will tell why it will not go up. Also, the output will include the update totals for the adds, drops, updates or changes. Optional output is a back-up file or a file of the recently updated transactions.

Information Retrieval

The mainline program is the "calling program" for retrieval programs. The specifications for the mainline program indicate the sets (files) to be selected and the subsets (data elements) of each file. A general retrieval program SELEK will be implemented, which will select entire records based on satisfaction of key or data element criteria. (Example: "all assistant professors in the social science department.") The Wabash SELEX program will select only those portions of a record required for output and reblock them. Output are the optionally sorted and selected records. Many frequently used programs will use predefined selection routines.

Computation

The computation module will also be called by the mainline program. Some frequently used programs will use pre-defined computation algorithms. These include tallies, totals, statistical subroutines, plotting, RRPM 1.6 simulation algorithms, enrollment projection models, grade point average, computations, etc.

Optional Post Sort

The optional post-sort will also use the RSTS Sort Package. Input will be the retrieved computed file and the output will be the sort and/or merged file. Output can be on a temporary disk or tape file or another temporary file to be used for reporting purposes.

Report Generation

The primary purpose of the system is to generate reports. In the anlysis phase it was determined that the system will eventually produce over 100 reports, which is adequate justification for developing a report generator. the report generator REPRT will employ the attribute file. The reports will also be called during the mainline program. By using the attribute file for formatting, the print spacing can be done with only insertion of blanks. Programs such as TIAA report, which do not print any original data element, will generally be sorted as sub-routines.

CODASYL DATA BASE REPORT

James Gilliam
B. F. Goodrich Chemical Company
Cleveland, Ohio

INTRODUCTION

The Integrated Data Base Management System (IDMS) is a subset of the April 1971 CODASYL Data Base Task Group Language specification. IDMS is designed to provide database facilities for ANS COBOL programs and any other host language which supports a CALL statement or equivalent, operating in an IBM 360 or 370 OS/DOS/VS environment. It provides Data Manipulation Language (DML) statements which are used to store, retrieve and manipulate data stored on random access devices. Data manipulation statements may be included anywhere within the procedure coding of an application program. Before compilation, a preprocessor validates and converts all data manipulation statements into CALL statements. Working Storage is used to establish database record I/O areas and communications between the user and IDMS. The IBM Basic Direct Access Method (BDAM) is used for physical I/O between IDMS and the direct access storage device.

IDMS provides separate language facilities for description of data and the manipulation of data. This separation removes the data description function from the scope of application programs. This separation of data description facilities allows integration of all data and data relations into a database which is common to all application programs which use it.

DATA DESCRIPTION

IDMS supports a network or hierarchical type of data structure. This facility permits user definition of structures which are most suitable to the applications which operate upon the data.

Data description within IDMS is done to provide a complete description of the database and to describe portions of a database to one or more separate application programs. While there is only one complete description for a database, there may be any number of sub-descriptions, each describing a specific combination of record segments, records, sets and areas which apply to a given application or program. This separation means that application projects do not have to know the whole database but only the particular part that concerns them, and can be insulated from changes to any but their own part.

Data description is accomplished with a two-part language called the IDMS Data Description Language (DDL). This language is used for both the description and sub-descriptions by an installation's database administration staff, which would be responsible for the design, maintenance and protection of the database.

The complete description is referred to as the Schema, and the part of the Data Description Language used to describe the Schema is called the SCHEMA DDL. The sub-descriptions are referred to as the subschema, and are described with the SUBSCHEMA DDL. Both the SCHEMA DDL and the SUBSCHEMA DDL are implementations of the specifications of the April 1971 report of the CODASYL Data Base Task Group Report.

DATA MANIPULATION

Data manipulation statements may be included anywhere within the procedure coding of a COBOL or other language application program. The data manipulation statements can be grouped into three categories: control, retrieval, and modification. They are summarized in Figure 1.

Control statements are used to establish access to a portion of a database. The OPEN AREA statement announces the user's intention to begin processing within the area specified as part of the statement. The corresponding CLOSE AREA statement announces the end of processing in the named area. An area is a named subdivision of contiguous addressable storage within the database.

Retrieval statements are primarily concerned with locating information in the database and making it available to the program in Working Storage. The IDMS FIND, GET, OBTAIN and MOVE CURRENCY statements are included in this category. The FIND statement locates a record occurrence in the database which satisfies the record-selection-expression part of the FIND statement. The GET statement causes a movement of all data items in the record which was most recently located into Working Storage. The OBTAIN statement is a combination of FIND and GET. The MOVE CURRENCY statement causes movement of the database-key associated with the most recently located record of a set, area, or record type into a named location in Working Storage. A database-key is used as a unique identifier of each record occurrence in the database.

Modification statements result in a change to the contents of the database. Changes include the addition of new data, modification of existing data by replacement of existing data item values, or deletion of data which currently exists in the database. The STORE statement uses data established by the user in Working Storage to create a new record occurrence in the database. The MODIFY statement is used to change the data content of an existing record in the database. The INSERT and REMOVE statements cause a change in the set relationship of an existing record occurrence in the database. The DELETE statement causes an existing record occurrence to be removed from the database. The space previously used to store a deleted record becomes available for storage of new record occurrence.

The application programmer can only access a database through a subschema which has to be predefined for him by the database administration staff. A subschema is made available to a program by an INVOKE declarative statement. The DML processor supplies record type descriptions from the subschema invoked into the working area of the user's program. The record segments, records, sets, and areas defined in the subschema are the only data that can be manipulated by the data manipulation statements of the application program; anything not included in the subschema cannot be touched. This provides a measure of data privacy and data independence at the application program level. The imperative data manipulation statements (for control, retrieval, and modification) and the declarative statement INVOKE are referred to collectively as the Data Manipulation Language (DML).

INTEGRATED DATABASE MANAGEMENT SYSTEM

```
CONTROL STATEMENTS
    SCHEMA SECTION.
    INVOKE SUBSCHEMA subschema-name OF schema-name.

                                 { RETRIEVAL           }
    OPEN ALL AREAS USAGE-MODE IS { EXCLUSIVE RETRIEVAL } .
                                 { EXCLUSIVE UPDATE    }

    CLOSE ALL AREAS .

  FORMAT 1
    IF set-name SET [NOT] EMPTY GO TO procedure-name.
  FORMAT 2
    IF RECORD [NOT] MEMBER OF set-name SET GO TO procedure-name.

MODIFICATION STATEMENTS
                                 [ ONLY      ]
    DELETE record-name RECORD    [ SELECTIVE ] .
                                 [ ALL       ]

    INSERT  record-name RECORD INTO set-name SET .

    MODIFY  record-name RECORD .

    REMOVE  record-name RECORD FROM set-name SET .

    STORE   record-name RECORD .
```

```
RETRIEVAL STATEMENTS
  FORMAT 1 (Direct retrieval using database key)
    { FIND   }
    { OBTAIN }  record-name RECORD USING identifier.

  FORMAT 2
                          { record-name  RECORD }
    { FIND   }            { set-name     SET    }
    { OBTAIN } CURRENT OF { area-name    AREA   } .
                          { RUN-UNIT            }

  FORMAT 3
               { NEXT  }
    { FIND   } { PRIOR }                          { set-name  SET  }
    { OBTAIN } { FIRST }  [record-name] RECORD OF { area-name AREA } .
               { LAST  }

  FORMAT 4
    { FIND   }
    { OBTAIN }  OWNER RECORD OF set-name SET .

  FORMAT 5 (CALC retrieval using CALC data-item)
    { FIND   }
    { OBTAIN } [NEXT DUPLICATE] record-name RECORD .

  FORMAT 6 (Retrieval in ordered set using control data-item)
    { FIND   }
    { OBTAIN }  record-name RECORD VIA CURRENT OF set-name SET USING identifier .

    GET record-name RECORD .

                                { RUN-UNIT TO identifier              }
    MOVE CURRENCY STATUS FOR    { record-name RECORD TO identifier    } .
                                { area-name AREA TO identifier        }
                                { set-name SET TO identifier          }
```

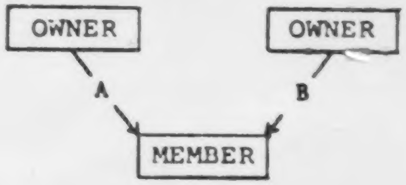
Figure 1

LOGICAL DATA RELATIONSHIPS

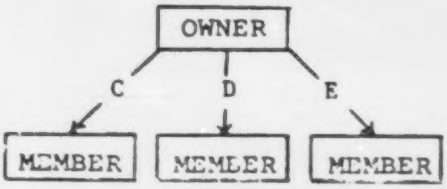
Perhaps the most important question of any DBMS is what logical data relationships can it support. IDMS gives the user far more choices of relationship types than any other competitive system. Logical relationships between two or more record types are established by the set mechanism. The set is the building block which allows various data structures to be built.

There are seven basic rules for the formation of IDMS set relationships between record types:

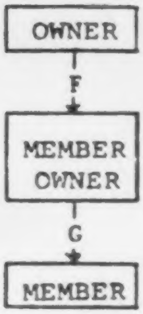
1. Any record may participate as a member in one or more sets.


OWNER
OWNER
A
B
MEMBER

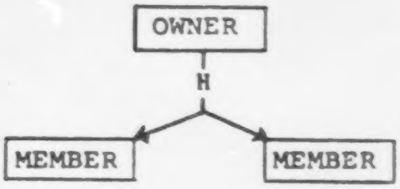
2. Any record may be the owner of one or more sets.


OWNER
C
D
E
MEMBER
MEMBER
MEMBER

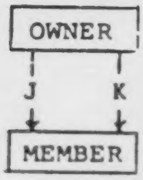
3. Any record may participate as a member in any number of sets, and also be an owner of one or more sets.


OWNER
F
MEMBER
OWNER
G
MEMBER

4. A set may have only one record type as its owner but may have one or more record types as members.

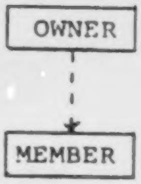

OWNER
H
MEMBER
MEMBER

5. Any number of set relationships may exist between two record types.



6. A record type may exist without set participation, that is as neither a member or owner in any set.

7. Any record type may be defined as an optional member of a set. Participation of a record occurrence is established or deleted based on execution of a procedure within a user program.



RECORD PLACEMENT CONTROL

General placement of a record type in the database is achieved by specification of an area in which all occurrences of the record type will be stored. Additional storage placement control within an area is achieved by one of the following location mode options:

- The DIRECT option allows maximum user control over logical storage location of a record occurrence within an area. In IDMS, the user can supply a "suggested" database-key before a record is to be stored. If the suggested database-key is available, it will be assigned to the record occurrence. If unavailable, the next available database-key will be assigned by the DBMS.
- The CALCULATED (CALC) option stores the record occurrence based on the value of one or more data-items within the record. A mathematical transform algorithm uses the item values to produce a logical storage position within the area associated with the record.
- The VIA option specifies that record occurrences will be accessed primarily as members of a named set and stores the records as near as possible to the owner occurrence.

These storage options allow the user to determine the number of entry points into the database and control access performance.

SET ORDER

Many options are provided for ordering member record occurrences within a set. The intra-set position a new record occurrence will occupy when stored in the database may be specified as the first record (top of a push-down list), the last record (bottom of a push-up list), or a position immediately before or after a record occurrence established by the user program. A set may also be ordered by a record type if more than one record type is included in a set as well as by a database-key.

Each record occurrence may be automatically maintained in ascending or descending order based on the value of one or more data items included within the record. The logical ordering of member record occurrences in a set is independent of the physical placement of the records. Furthermore, and most important, the same record occurrence may participate in any number of different sets, each with different ordering criteria.

SET MEMBERSHIP

The type of membership may also be different in each set. Set memberships for a record type may be mandatory or optional depending on whether the user program is permitted to insert, or remove, an occurrence of this record type from the set. A mandatory set membership means that a record occurrence is a permanent member of the set as long as the record is present in the database. An optional set membership allows a record occurrence to be removed from a set without being deleted from the database. Set membership may also be specified as an automatic function performed by the DBMS when a record occurrence is stored in the database, or as a manual function to be performed by the user program.

SET LINKAGE OPTIONS

Members of a set are automatically linked in the forward directions (next pointers). The user may optionally request that the set be linked in the reverse direction (prior pointers) so the chain may be processed in either direction. The user may also specify that the owner pointers be carried in the member record occurrences so that the owner record occurrence is known and can be accessed without following the chain to its head. Therefore, the options for set linkages are to have next pointers, next and prior pointers, next and owner pointers, or next, prior and owner pointers.

The DDL schema syntax is summarized in Figure 2.

SCHEMA DDL SYNTAX

```
SCHEMA DESCRIPTION.
SCHEMA NAME IS schema-name
                   [RANGE IS integer-1 THRU integer-2].
[AUTHOR.           [comments]... .]
[DATE.             date-entry.]
[INSTALLATION.     [comments-... .]
[REMARKS.          [comments]... .]

AREA DESCRIPTION.
        AREA NAME IS area-name
                   RANGE IS integer-1 THRU integer-2.
RECORD DESCRIPTION.
      RECORD NAME IS record-name.
      RECORD ID IS integer.
      LOCATION MODE IS
          { DIRECT                                                }
          { CALC USING identifier                                 }
          {         DUPLICATES ARE { FIRST                }       }.
          {                        { LAST                 }       }
          { VIA set-name SET       { NOT ALLOWED          }       }
          {         [DISPLACEMENT integer PAGES]                  }
      WITHIN area-name area
                   [FROM integer-1 THRU integer-2].
      level-number { data-name-1 }
                   { FILLER      }
                   [REDEFINES data-name-2]
                   [                 { DISPLAY                          } ]
                   [ [USAGE IS]      { COMPUTATIONAL     (COMP)         } ]
                   [                 { COMPUTATIONAL-1   (COMP-1)       } ]
                   [                 { COMPUTATIONAL-2   (COMP-2)       } ]
                   [                 { COMPUTATIONAL-3   (COMP-3)       } ]
                   [SYNCHRONIZED (SYNC)]
                   [PICTURE (PIC) IS character-string]
                   [VALUE IS literal]
                   [OCCURS integer TIMES]
                   [INDEXED BY index-name].
```

```
SET DESCRIPTION.
      [ SET NAME IS set-name.                                           ]
      [                { FIRST  }                                       ]
      [ ORDER IS       { LAST   }.                                      ]
      [                { NEXT   }                                       ]
      [                { PRIOR  }                                       ]
      [                { SORTED }                                       ]
      [ MODE IS CHAIN [LINKED TO PRIOR].                                ]
      [ OWNER IS record-name                                            ]
      [           NEXT DBKEY POSITION IS integer-1                      ]
      [           [PRIOR DBKEY POSITION IS integer-2].                  ]
      [ MEMBER IS record-name                                           ]
      [           { MANDATORY } { AUTOMATIC }                           ]
      [           { OPTIONAL  } { MANUAL    }                           ]
      [           [LINKED TO OWNER]                                     ]
      [           NEXT DBKEY POSITION IS integer-1                      ]
      [           [PRIOR DBKEY POSITION IS integer-2]                   ]
      [           [OWNER DBKEY POSITION IS integer-3]                   ]
      [           [ { ASCENDING  } KEY IS identifier          ]         ]
      [           [ { DESCENDING }                            ]         ]
      [           [                           { FIRST       } ].        ]
      [           [          DUPLICATES ARE   { LAST        } ]         ]
      [           [                           { NOT ALLOWED } ]         ]
```

SUBSCHEMA DDL SYNTAX

```
SUBSCHEMA IDENTIFICATION DIVISION.
SUBSCHEMA NAME IS subschema-name
                         OF SCHEMA NAME schema-name.
[AUTHOR.                 [comments]... .]
[DATE.                   [comments]... .]
[INSTALLATION.           [comments]... .]
[REMARKS.                [comments]... .]

SUBSCHEMA DATA DIVISION.
        AREA SECTION.
        COPY area-name AREA.
        RECORD SECTION.
        COPY record-name RECORD.
        SET SECTION.
        [COPY set-name SET.]
```

FIGURE 2

D615	Decimal Precision - Number of positions to right of decimal point within the size specified in D605.
D620	Expansion Code Z = Replace any blanks with zeros. R = Right zero fill. More data to left boundary and replace blanks with zeros. L = Left zero fill. More data to right boundary and replace blanks with zeros. D = Date field conversion from MO-DA-YR to YR-MO-DA.
D625	Element Mnemonic - Name used in all COBOL programs.
D630	Description - Element description.
D635	Required - If Y (Yes), this element must be on the transaction. It cannot be blank.
D640	Table Value Edit Required - If Y (Yes), an edit will be performed for specific value or range of values. These values are specified in "VALU" entries in TABL.
D645	Data Element - Master Edit - If a data element number is entered here, there must be a record in the corresponding Data Base Master File whose Key is the contents of this field.
D650	Master Status Edit Required - If a Y is entered, there will be a TABL file record which contains an element number for a master file, a value and a relational operator. The contents of the element are evaluated, based on the relational operator (=, ≠, >, <) and the value.
D655	Variable Status Edit Required - If a Y is entered, there will be a TABL file record which contains an element number for a variable file, a value and a relational operator. The contents of the element for each record on the specified linkage path are evaluated based on the relational operator and the value.

D660	Relative Location Within Sort Key - This will indicate the relative position of this item in the Sort Key. A 50-character user Sort Key may be defined as a part of the system and data.
D665	Edit Required Code - If Y, the transaction element will be edited.

As I have tried to emphasize throughout this paper, a Data Dictionary System need not have all of the preceding components, but without some of these essential capabilities, I would dare say that any effort to implement a Data Base Management System would be much more lengthy and costly than if some time were given to designing a dictionary prior to implementation.

In closing, I would reiterate a statement made throughout this paper that there is no right way to design a Data Dictionary System and that the specific design is related to local conditions as is the case with any system or program definition. The important point to consider is that consideration of the design of such a system is a first step before the development of applications or even defining the Data Base.

COBOL/DML COMPILATION

The user must identify a predefined subschema in the SUBSCHEMA SECTION of the Data Division. The subschema allows the user to manipulate only the data-items, records, sets and areas included in the subschema. The DML statements may appear anywhere in the Procedure Division. From a programming viewpoint, DML statements may be considered as an extension to the COBOL Language. This means that they may be placed at the appropriate point in any procedure and may be surrounded by other COBOL statements.

The compilation process is illustrated in Figure 3 . The IDMS DML processor reads the COBOL/DML source statements and searches within the Data Division for the Subschema Section to obtain the name of the subschema specified in the INVOKE statement. All COBOL (or non-DML) statements are copied directly to output. Once the subschema name has been verified, the DML processor obtains the names of all valid records, sets, and areas. The relationships between records, sets, and areas are also obtained from subschema information stored in the database description. These are used to validate DML statements in the Procedure Division. The DML processor will establish an 01 level record entry followed by selected data-item entries in Working Storage for each record included in the subschema. In addition, system control and communication data items will be included following the last record description.

The DML processor will insert a group of CALL statements immediately following the PROCEDURE DIVISION statement to supply the necessary Working Storage location of records and data items to the object subschema interface control routine. Each DML command is validated for correct syntax and usage of record-set-area relationships. If the DML statement is correct, the processor will convert the DML statement into a COBOL NOTE and generate an appropriate COBOL CALL statement to the subschema interface routine.

All DML errors detected by the DML processor will be displayed with the source statement which was in error. When input of COBOL/DML source is complete, statements output by the DML processor along with all other COBOL statements will be directed to the COBOL compiler.

OTHER HOST LANGUAGE/DML COMPILATION

IDMS may be used with any host language (e.g. FORTRAN, PL/I, BAL) that supports a CALL macro statement, or equivalent. Since the DML processor is available only for ANS COBOL, the user communicates with IDMS from other host languages by passing arguments through CALL statements. This CALL statement language is straightforward and easy to use and may even be used directly in COBOL if desired.

IDMS COMPILATION

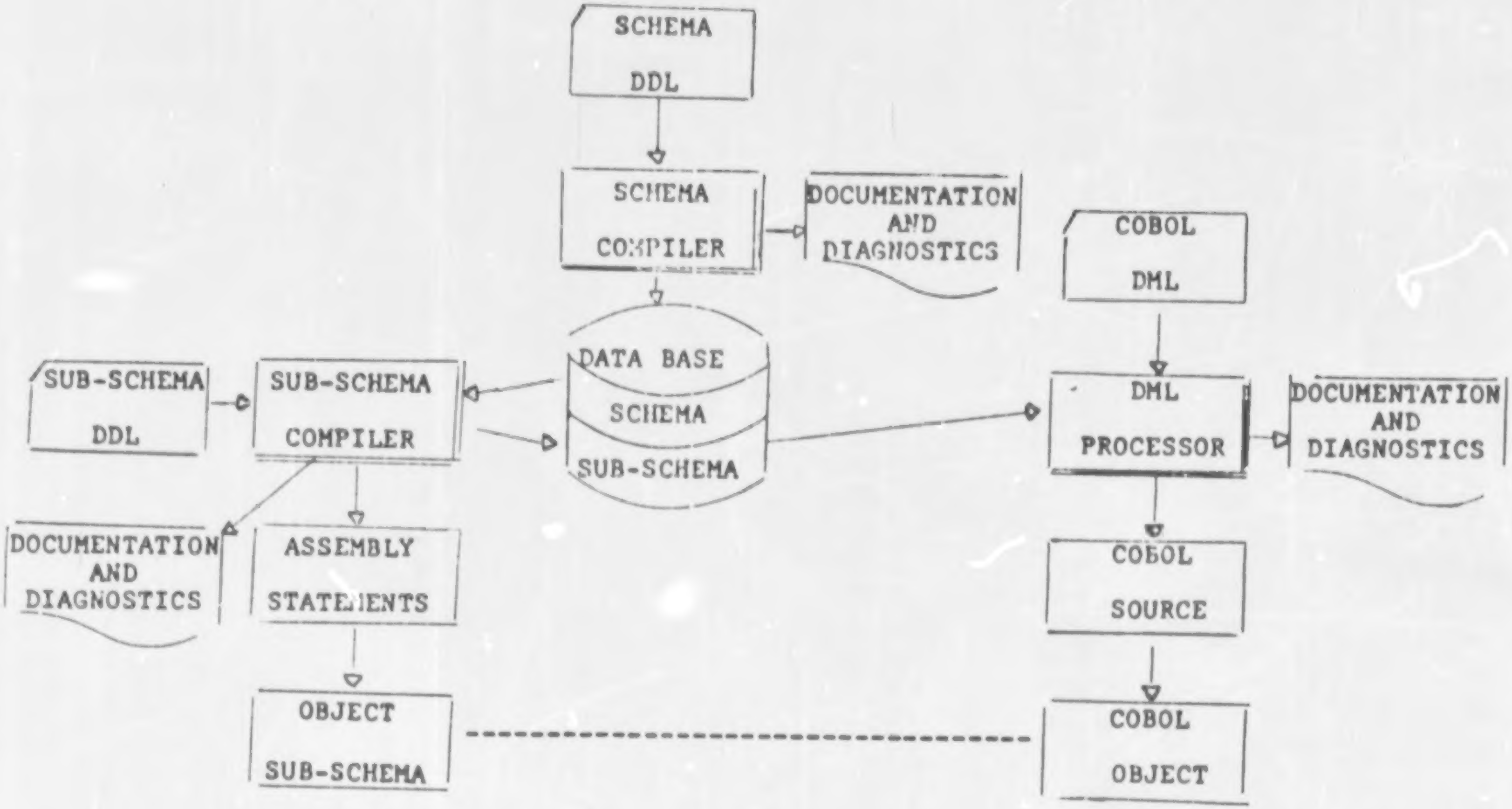

SCHEMA DDL
SCHEMA COMPILER
DOCUMENTATION AND DIAGNOSTICS
COBOL DML
DATA BASE
SCHEMA
SUB-SCHEMA
SUB-SCHEMA DDL
SUB-SCHEMA COMPILER
DML PROCESSOR
DOCUMENTATION AND DIAGNOSTICS
DOCUMENTATION AND DIAGNOSTICS
ASSEMBLY STATEMENTS
COBOL SOURCE
OBJECT SUB-SCHEMA
COBOL OBJECT

Figure 3

PROGRAMMING REQUIREMENTS

From a programming viewpoint, the IDMS input/output area for the database resides in Working Storage. Each record included in the subschema is automatically included in Working Storage by the DML processor as an 01 record level entry followed by the statements which describe the name, picture attribute and usage of each data item included in the subschema record. Only those segments of the physical record defined by the subschema are transferred to the user program. Input of a record from the database always appears in its like-named area in Working Storage. Storage (or output) of a record to the database requires the movement of data from various locations in the user's program to each of the data-items described for the specific record in Working Storage. Once this is completed by user procedure statements, the record is then physically moved from Working Storage into the database using the STORE statement.

The user is responsible for initializing all data-items required to successfully execute a DML statement and must ensure that the data is correct. IDMS has extensive object error diagnostic facilities and will update the error status following every DML statement in working area of the user's program. The user must examine the error status following each DML statement to determine the action taken by the system in response to a DML statement. In summary, three operations are required each time an access to the database is required:

Initialization of data-items as required by the DML statement to be executed;

DML statement to initiate a database operation;

Error checking to determine the outcome of the preceding DML statement.

Before a programmer can write any DML statements, the names of all areas, records, sets, and data items must be known along with their characteristics. The existence of data structures present in the database but not included in the subschema may affect the scope of the DML statements which can be used with the subschema. This subschema information and any DML restrictions must be documented by the database administration staff and understood by the systems analyst and programmer before program specification and implementation can be accomplished.

EXECUTION OF OBJECT COBOL/DML PROGRAMS

The following will explain the operations which take place when a DML statement is executed. The numbers in parentheses refer to figure 4.

(A) A DML statement appears in the object program as a call to the subschema interface routine (1). The call identifies the type of database service desired and any additional information such as record-name, set-name and area-name required to properly interpret the call.

(B) The subschema interface routine analyzes the call using information stored in the object subschema (2). If the call requires the use of IDMS, the user-supplied information is augmented by information from the object subschema before control is passed to IDMS (3).

(C) IDMS performs the requested database service using information supplied by the interface routine. The operation which is performed depends upon the type of DML statement executed. In the event of a call to locate a record (FIND or OBTAIN statement), IDMS will look in the system buffers to see if the requested record occurrence is present. If the record is not in the system buffers, a request will be made to the operating system to input a database page from the direct access storage to the IDMS system buffers (4,5,6). No input occurs if the record is already present in the system buffers.

(D) Any changes to the database are recorded on the log file (7).

(E) IDMS will perform the requested database service using additional information contained in the object subschema. The object subschema contains a representation of the data structure, record placement control, record characteristics, set characteristics, currency status, database operation statistics, and constraints on DML operations. In general, the object subschema controls the operations and access of the database for each program which invokes it.

(F) Whenever a specified record occurrence is located by IDMS, the database-key and other system information related to the record is moved from the system buffer to locations within the object subschema (8). This information represents the currency status of the area, sets, and record-type of the record occurrence which has been located.

(G) If the call to IDMS specified movement of the contents of a record to the user working area (GET or OBTAIN statement), data will be moved from a system buffer to a specified record area in the user working area (9). Data movement from the user working area to the system buffers will occur in response to a STORE or MODIFY statement.

(H) IDMS returns to the subschema interface routine with an indication of the success or failure of the database service which was requested in step C (10).

(I) The interface routine moves status information regarding the outcome of the DML statement executed in step A to locations within the working area of the program (11).

(J) Control is returned to the user's program at the statement following the DML statement just executed (12).

(K) The user must determine the status of the previous DML statement by examining the contents of the system status data items in his working area (13). For example, if a FIND statement was just executed, the contents of a data item named ERROR-STATUS would indicate whether the specified record occurrence was located or not. If the status condition indicates that the service requested was completed successfully, the user would then access his working area as needed (14).

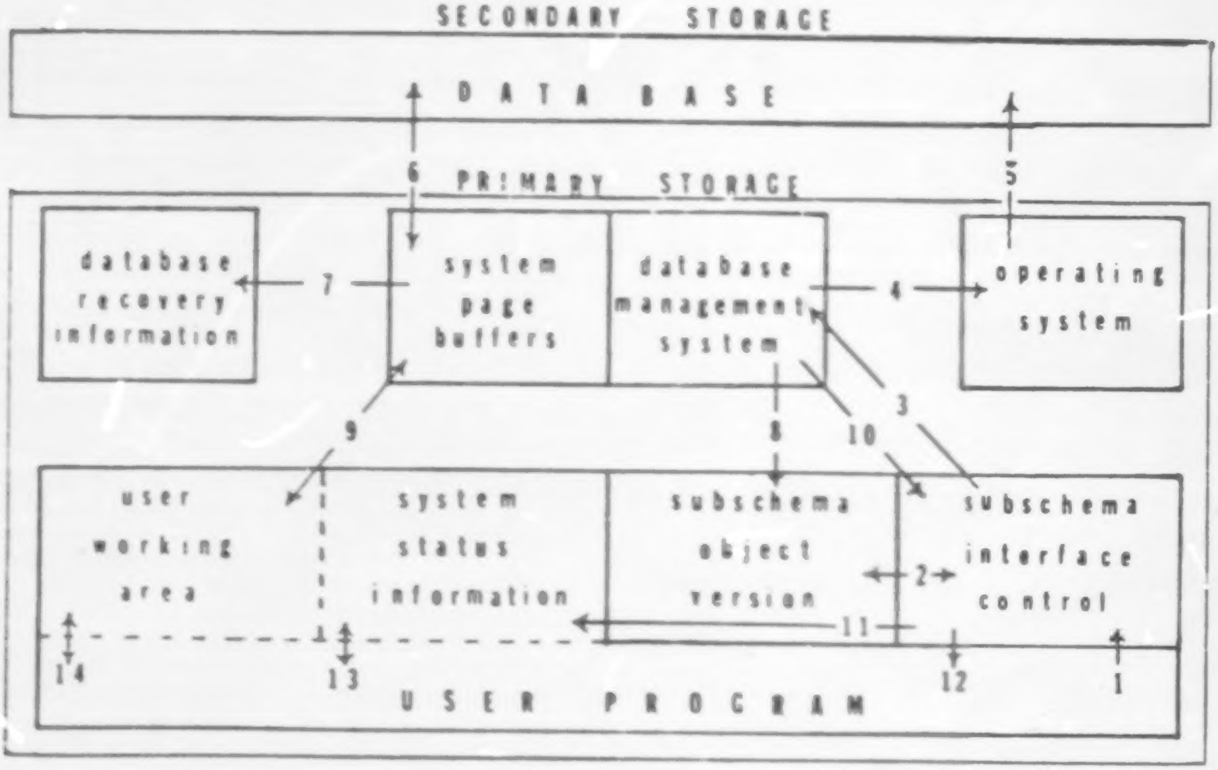

SECONDARY STORAGE
DATA BASE
PRIMARY STORAGE
6
5
database recovery information
7
system page buffers
database management system
4
operating system
3
9
8
10
user working area
system status information
subschema object version
2
subschema interface control
11
14
13
12
1
USER PROGRAM

Figure 4

IDMS FEATURES

IDMS is a subset implementation of the April '71 CODASYL Data Base Task Group Language Specifications.

Designed as a language extension to ANS COBOL.

May be used with any host language that supports the CALL statement or equivalent.

Permits the user to specify a variety of data structures including network, hierarchical, "bill of material", or a combination.

Allows any number of user-specified entry points into the database.

Provides data/program independence.

Provides control over physical placement of records to optimize performance.

Eliminates need for designated overflow storage.

Eliminates the need for periodic reorganization of the database.

Includes extensive error analysis of all database operations.

Provides utilities to monitor database performance and storage density.

Provides database rollback/recovery facilities to recovery integrity of the database following either a software or hardware failure.

Operates under IBM 360/370 OS/DOS/VS.

Disk access is through standard use of BDAM.

Any direct access device supported by BSAM and BDAM may be used including 3330's.

Audit trail information is output to any sequential file using standard BSAM.

Memory required by the IDMS object code, buffers, interface routines and object subschema is approximately 50K.

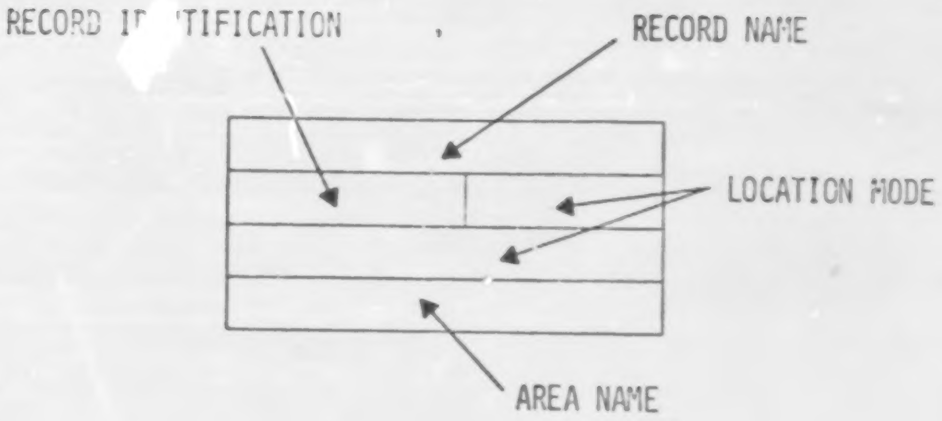
IDMS has been in production use since January 3, 1972.

IDMS STRUCTURAL DOCUMENTATION

IDMS logical structure is the same as its physical structure. This makes it easy to document and learn. Figure 5 explains the Documentation for a RECORD TYPE. Figure 6 explains the Documentation for SET TYPES. Figure 7 shows a subschema example that a data administrator would give the programmer on documentation of the logical and physical structure he is working with. The documentation method is used for the schema by the data base administrator as well as the subschema which is a logical subset of the schema.

RECORD REPRESENTATION

GENERAL FORMAT

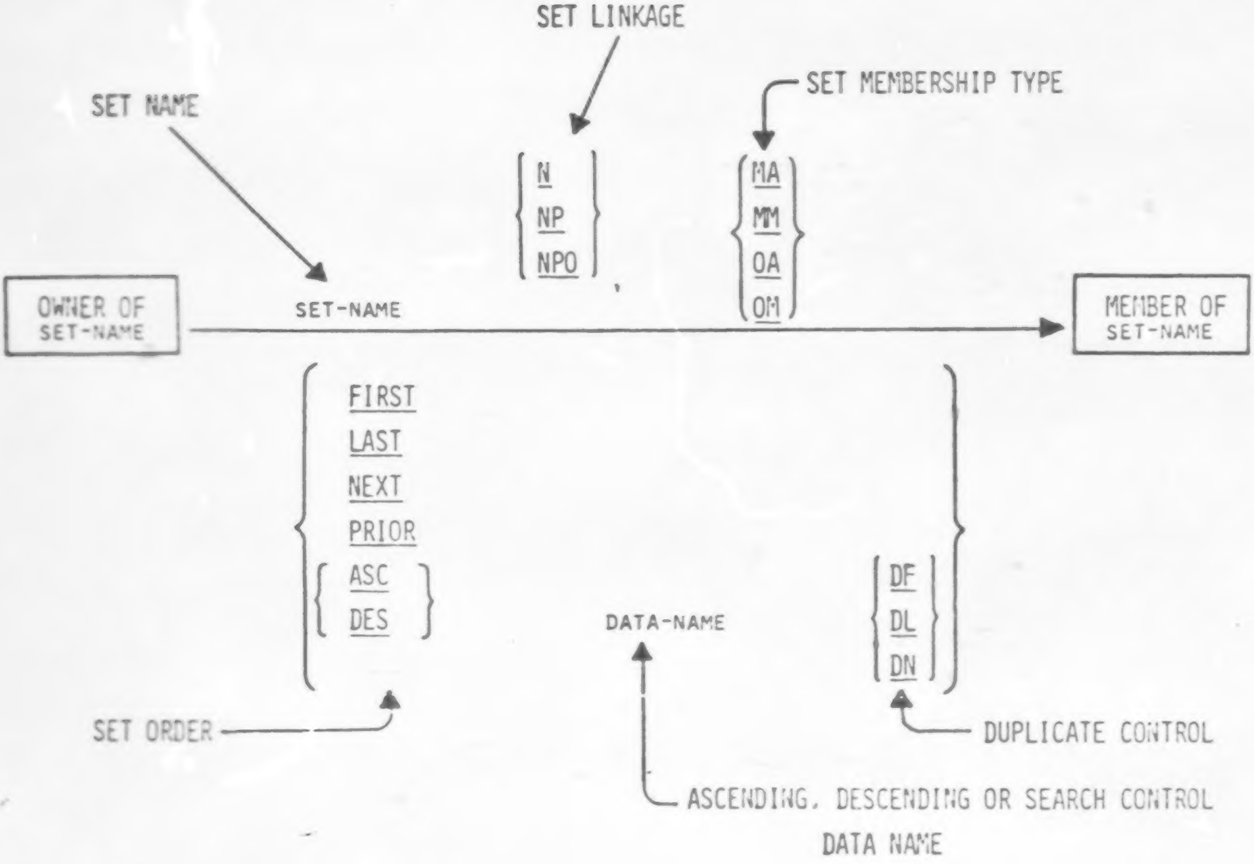

RECORD IDENTIFICATION
RECORD NAME
LOCATION MODE
AREA NAME

EXAMPLE RECORD REPRESENTATION

CUSTOMER	
611	CALC
CUST-NO	
TOPSY	

FIGURE 5

SET ATTRIBUTES


SET LINKAGE
SET NAME
SET MEMBERSHIP TYPE
N
NP
NPO
MA
MM
OA
OM
OWNER OF SET-NAME
SET-NAME
MEMBER OF SET-NAME
FIRST
LAST
NEXT
PRIOR
ASC
DES
DATA-NAME
DF
DL
DN
SET ORDER
DUPLICATE CONTROL
ASCENDING, DESCENDING OR SEARCH CONTROL DATA NAME

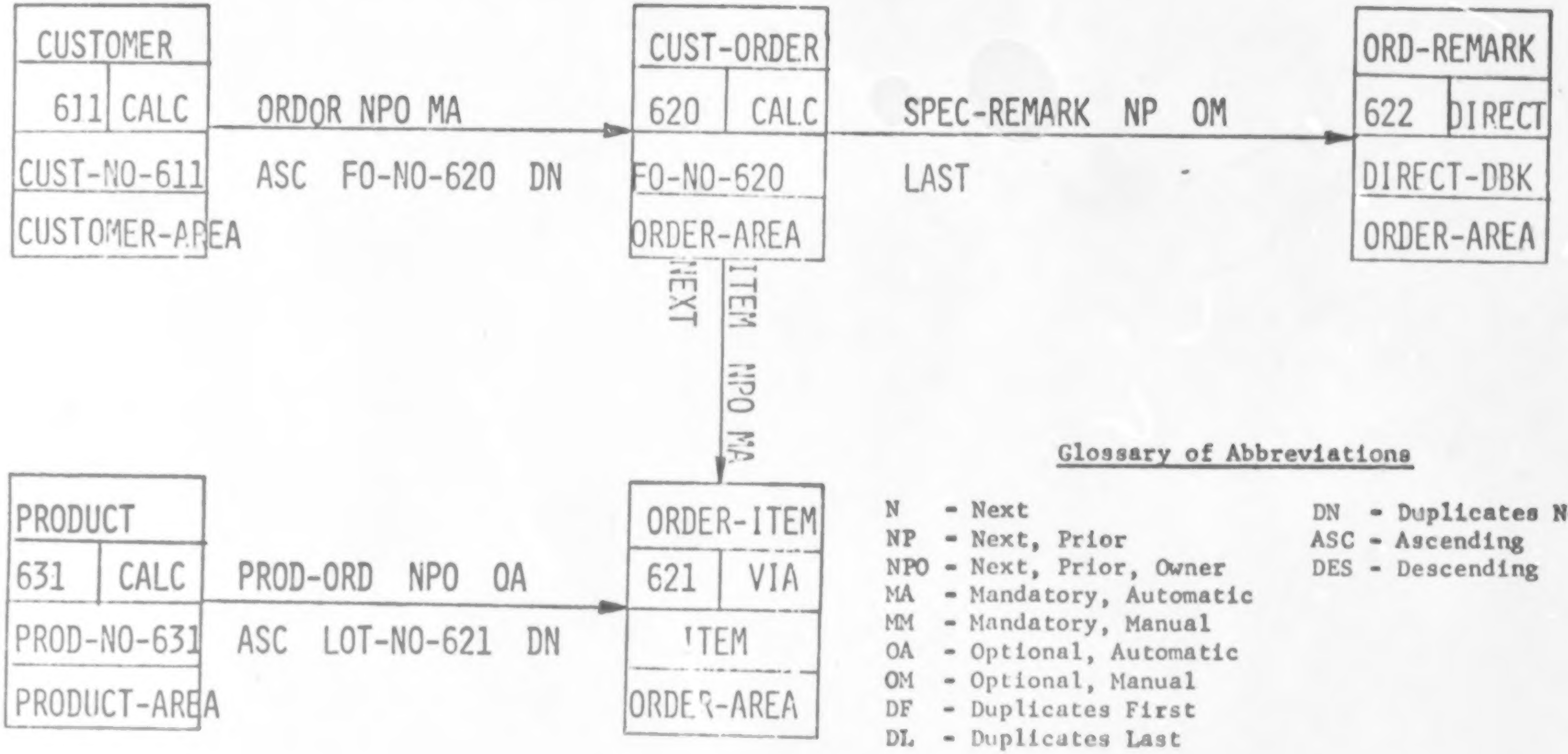
ABBREVIATIONS

ASC - Ascending order based on data-name
DES - Descending order based on data-name
DF - Duplicates First
DL - Duplicates Last
DN - Duplicates not allowed
MA - Mandatory Automatic
MM - Mandatory Manual
N - Next
NP - Next and Prior
NPO - Next, Prior and Owner
OA - Optional Automatic
OM - Optional Manual

FIGURE 6

DATA STRUCTURE DIAGRAM

SUBSCHEMA NAME OPENORD
SCHEMA NAME BFGCHEM


CUSTOMER
611 CALC
CUST-NO-611
CUSTOMER-AREA
ORDOR NPO MA
ASC FO-NO-620 DN
CUST-ORDER
620 CALC
FO-NO-620
ORDER-AREA
SPEC-REMARK NP OM
LAST
ORD-REMARK
622 DIRECT
DIRECT-DBK
ORDER-AREA
ITEM NPO MA
NEXT
PRODUCT
631 CALC
PROD-NO-631
PRODUCT-AREA
PROD-ORD NPO OA
ASC LOT-NO-621 DN
ORDER-ITEM
621 VIA
ITEM
ORDER-AREA

Glossary of Abbreviations

N	- Next	DN	- Duplicates No
NP	- Next, Prior	ASC	- Ascending
NPO	- Next, Prior, Owner	DES	- Descending
MA	- Mandatory, Automatic		
MM	- Mandatory, Manual		
OA	- Optional, Automatic		
OM	- Optional, Manual		
DF	- Duplicates First		
DL	- Duplicates Last		

FIGURE 7

DATA BASE ADMINISTRATION: THEORY INTO PRACTICE

Jane N. Ryland
Division of Management Services
Virginia Department of Community Colleges

DATA BASE ADMINISTRATION :

THEORY INTO PRACTICE

With the growth of popularity of data base management systems, there has been a corrolative development of a personnel function known as data base administration. The implementation of a data base management system, with its inherent integration of previously independent activities, forces a much more thorough look at the overall objectives or goals of the enterprise, and requires a coordination of the developmental procedures,which previously was not necessary. When an application area was developed under the traditional isolated approach, the developers needed to be aware of only that single area and the constraints of the environment in which the application would operate. With an integrated system, it is essential that some individual or group display an overall awareness of the system, both as it exists at that time, and in light of planned future development.

DEFINITION OF THE FUNCTION

The increased cognizance of the importance of data as a resource has generated substantial interest in the management of this data via evolving data base technology. Several attempts have been made to define the function of data base administration, the most significant of which is probably a document prepared by the Data Base Administration Project in GUIDE, entitled "The Data Base Administrator." Generally, such topics as responsibilities of the data base administrator, the tools which he has at his disposal, his relationship with the various functional areas of the enterprise, and staffing considerations are included in these definitions.

POSITION IN THE ORGANIZATION

The data base administrator can conceivably function at several levels within the organization. General opinion seems to favor as high a position as possible. Indeed, a position external to the computer facility itself and on an equal or higher level can present excellent opportunities for innovation and control. However, this is not essential to the successful execution of the duties of the position. From experience it is determined that the job can be effectively handled by a person reporting to the manager responsible for systems development, as long as the individual personalities are compatible, and both are dedicated to data management concepts. With this particular method of organization, the data base administrator is advantageously placed, since the majority of his work involves systems development personnel. However, the goals of the systems development manager are usually more oriented to the speedy implementation of systems, and conflicts can arise when the data base administrator imposes delaying controls. Probably the best organizational position that can be occupied by the data base administrator would be one associated with the data processing facility, reporting directly to the chief executive of that facility. This seems to provide the best combination of independence from and close association with the key personnel involved in data management.

AUTHORITY

The organizational position of the data base administrator can directly affect his authority in the organization. Here again, there are several alternatives. In defining a data base in terms of its contents, interrelationships of individual data elements, and the authority to access data in a particular manner, the data base administrator can have ultimate responsibility for these decisions, or the responsibility can reside with some other individual or group. In evaluating these alternatives, it seems preferable to elect the latter. The data base administrator should be primarily a technician and should not have to formulate policy for the organization in regard to overall objectives and the right or need to store, interrelate, and access specific information. This is especially true for a data base administrator associated with the data processing facility whose charge is that of a service utility to the organization. In a small organization, in which there is an absence of an individual or group qualified to make such decisions, the data base administrator can perhaps fulfill a double role. In this case, it would be best to place the data base administrator outside of the data processing facility, sacrificing the better working environment for the added authority. One approach which can work successfully is the creation of a data base committee composed of reasonably high level executives whose responsibility it is to interpret the policy of the organization in regard to the data base. The data base administrator can participate on this committee, gaining the knowledge necessary to accurately reflect the nature of the organization in the data base, and providing the technical knowledge to properly direct the activities of the committee.

THREE CATEGORIES OF FUNCTION

Essentially, the work of the data base administrator can be categorized into three groups in terms of the time at which the work is performed. Before any actual systems are developed there is a considerable amount of preliminary work necessary to establish the basis for applications development. The data base administrator must play an integral part in this activity to insure that the proper environment is created. Next, as each applications area is integrated into the data management system, the data base administrator should be deeply involved in all work pertaining directly or indirectly to the data base. Finally, if and when new development is completed, the data base administrator must continually monitor the effectiveness of the data base system against the ever-changing functioning of the enterprise.

PREPARATION OF ENVIRONMENT

In the early stages of the development of a data base, the data base administrator spends a great deal of time in the preparation of the environment. He must insure that the capability exists to provide adequate physical storage, effective operational procedures, generalized utility programs, a workable and secure procedure for testing, security, backup and recovery procedures, a complete educational program, effective standards for programming and documentation, and an appropriate method for the user to interface with and thereby extract information from the data base. In performing these tasks, a division of responsibilities can be defined between the data base administrator and various individuals or groups. It is useful to consider alternate methods of dividing these responsibilities. For example, the data base administrator must work with the person responsible for systems analysis and programming to establish standards for programming and systems documentation which are compatible with the data base management technology being utilized. Usually, the systems and programming manager will have to realign his thinking regarding systems documentation, since the emphasis in data base systems is on the data base itself instead of the programs which make up the system. This redirection of emphasis must be reflected in the systems documentation. In addition, standards must be established for naming data bases, data sets, and data elements. It is important that users be involved in the naming process to insure that the names will be meaningful to non-data processing personnel. The data base administrator can either work directly with the users, or with the analysts responsible for each applications area who reflect the users' ideas. Once the naming standards have been established, the analysts can work with the users to assign the actual names to conform to the standards. The data base administrator will probably wish to review the selected names with the analysts. This method is preferable to one in which the data base administrator works with each user to assign individual names since the analyst is probably much more familiar with the applications area, and normally works closely with the user on all aspects of the system.

AD HOC COMMITTEES

Much of the work of the data base administrator in the early stages of data base development will involve the creation of a variety of documents which will provide "ground rules" for later applications implementation. Two problems often faced by the data base administrator at this time are the lack of a staff to perform many of these functions, and the necessity to educate the analysts, programmers, and operators in data base technology. A viable solution seems to be the organization of one or more ad hoc committees, composed of a broad range of computer personnel, to address specific needs. The data base administrator can meet with these groups to provide necessary direction. Some functions that can be documented effectively by such committees are operational procedures for the data base systems, procedures to be followed in the event of system failure or destruction of a portion of the data base, data base reorganization guidelines, program and system test procedures and definition of communication channels. The committee members should represent all areas of the computing center, and can thus establish workable and acceptable procedures for all. The opportunity to express opinions and participate in decisions regarding the environment in which they will be working can be quite beneficial from the standpoint of both education and personal satisfaction.

Depending on the size of the organization, the data base administrator can actually perform systems programming duties related to data base management, or he can divide this responsibility with systems programming personnel. The latter approach is probably preferable, since even in a small installation, there will be aspects of the total system not directly related to data management, such as the operating system itself, which will require a systems programmer. The systems programmer will thus already be familiar with procedures for generating and maintaining systems software, and can insure that the data management system interfaces properly with the complete systems environment. However, it will be necessary for the data base administrator to work closely with the systems programmer to define the parameters used in tailoring the data management system to meet the needs of the users. Once the initial system has been defined, procedures must be established to modify the system as new applications are developed and implemented.

DATA BASE DESIGN

Once the data base environment has been established, the data base administrator is heavily involved in the addition of each application area to the data base. Ideally, the data base administrator should act as a consultant to the analysts and programmers assisting with technical decisions of data base structure and organization, distributing memos on suggested techniques, approving names of data elements and data bases, and solving implementation problems. In this mode of operation, the analysts would perform much of the actual data base design. To optimize this design function, the committee concept can again be utilized. A data base design committee, composed of the data base administrator and all analysts who perform data base design, can review each newly designed portion of the data base. This allows data base design expertise to be more widespread in the organization, gaining the benefit of additional ideas and insuring that the analysts are aware of the most current state of the data base. The data base administrator would not work directly with the users if this method were employed. Although this mode of operation is recommended, it is probably not practical to implement it initially. Rather, it can be attained in an evolutionary manner. The data base administrator can at first assume all data base design duties, working directly with users. The data base design committee at this point will simply be a means of educating the analysts as each section of the data base is designed and reviewed. The data base administrator can conduct the meetings of the committee as if it were a tutorial, posing to the analysts the design problems and explaining the solutions. Gradually, the analysts can assume the design responsibilities and the data base administrator's role will more nearly approach that of a consultant.

EVALUATION

Once an application area has been integrated into the data base, the data base administrator is responsible for evaluating the effectiveness of the design. He must provide the tools by which the design can be evaluated, and in addition, he must establish a means of getting feedback from the users. The operation of the data base system itself can be evaluated by reviewing statistics which reflect response times and utilization of system resources. The data base administrator must be able to relate this information to systems definition parameters, and then work with the systems programmer to modify the system in an attempt to provide a more efficient environment. Data base design should be evaluated by testing alternative designs against actual programs. In addition, the use of various portions of the data base should be monitored by counting the number of times each portion is accessed. The data base can then be restructured to facilitate access to the more frequently used portions.

CONCLUSION

In summary, the function of data base administration can best be accomplished if the interfaces between the data base administrator and other groups within the organization are clearly delineated, and the direct interfaces between the users and the data base administrator are minimized. In addition, if a good deal of technical design and documentation is assumed by a wide range of computer center personnel, the data base administrator can devote more of his time to consultative and evaluative functions.

FEATURE ANALYSIS: REGISTRATION SCHEDULERS

Vinod Chachra
Assistant Professor
Computing Center
Virginia Polytechnic Institute and State University

INTRODUCTION

Computer Aided Registration (CAR), for the third time over the last decade, is going through the painful process of redesign. Technological developments in the area of data processing and the increasing pressures on university administrators to better utilize their resources has made many registration systems inadequate or obsolete. Results of a recent survey (partially summarized in the appendix) show that each university places a different set of demands on its registration system. These demands are diverse enough that it would be safe to say that almost every university using CAR has its own version of a CAR system. However, there is a certain conceptual unity underlying this diversity and it is this common factor that will be the subject of this paper. Also, an attempt will be made to show how a data management system (like IMS) can help resolve many problems associated with running a conventional tape oriented system.

COURSES OR SECTIONS

The very first step in creating a registration system is to decide if the students will be given the opportunity to request courses, or specific sections of courses. The form of input, section-choice or course-choice, determines the type and sequence of processing that follows. Most registration systems in existence today allow for section-choice. A notable exception is the Perdue Academic Student

Scheduling (PASS) system (1).

A registration system based on section-choice which automatically assigns an alternate open section of the requested course, if the requested section is full (or conflicting), is essentially equivalent to a system based on course-requests. A section request system however has the additional ability to take note of the students time preference and instructor preference. These preferences come into effect only if room is available in the requested sections, and the only way to assure this is to process the student before the section is full. The processing of the student before the secton is full can happen either naturally or by assigning some priority to the student and priority is assigned only after the student has demonstrated a need for it. Thus, after the requested sections begin to fill up the secton request system approaches the course request system. The PASS system though primarily a course-choice system, allows for the section-choice process but "only after the student has demonstrated a need for it" (1). This of course is equivalent to the assignment of priority in a section-choice system. Therefore, even though the section-choice system is preferred, there is only a little difference between the outcomes of the two systems. An interesting variation is a dual system (see OSCAR (9)) which allows the students to request either courses or sections. This system, if desired, permits the student to restrict his enrollment to a specific section of a course. If the requested section is unavailable then an alternate section of the same course is not assigned.

The material presented hereafter is general enough to be applicable to the three forms of input discussions above. However, for the sake of clarity, it is assumed that the registration system is in fact a section-choice system. In the sections that follow, some of the objectives of the registration system are defined. Registration systems are then classified into three categories. Characteristics of these categories are discussed to show how each measures up to objectives of the system. A technical section presents some concepts essential to registration schedulers.

PRIMARY OBJECTIVES

The primary objective of any registration system is to facilitate the enrollment of students into classes. This process would indeed be very simple if there were sufficient capacity to meet all the student's requests as requested. However, this is not always possible and sometimes not desirable. Considerations like teacher load, class room space or requirement of special facilities limit the capacities of some courses to a level far below the total demand for it. (This is an overflow situation and requires that a student enroll in an alternate course.)

It is not always desirable to meet all the requests made by students even though the university has the facilities to do so. Low enrollments that represent poor utilization of university facilities may warrant cancellation of the course entirely. In other situations it might be desirable to spread over a period of time the student demand for a given course. As an example of the latter, consider a case where 100 freshmen are required to take chemistry lab in their first year. Ninety of them request this lab in the first semester. The university every semester has the capability to teach two sections of chemistry lab, each with a maximum capacity of fifty-five. In this case even though the resources are there to enroll all ninety, it would be desirable to teach only a single section of maximum capacity of fifty-five each semester thereby forcing thirty-five of the students to take it next semester. (This again would require that a student enroll in an alternate course.) In view of the above, one might say that the primary objectives of a registration system are threefold:

a. To provide a convenient means by which the university can satisfy x percent of the total requests made by students. This generally implies that some y percent of the students get all the courses they requested (not necessarily at the exact time spans requested).

b. To provide a means by which (100-y) percent of the students can conveniently make alternate choices for only (100-x) percent of the total requests.

c. To have the ability to permit students to make changes to their schedules prompted by b above or by other unforseen circumstances (failing courses, changing curriculum, etc.)

A study of the schedulers in existence indicates that an acceptable level of x is between 90 and 95 percent and y is between 75 and 85 percent.

SECONDARY OBJECTIVES

There are many secondary objectives of registration scheduling systems. All of these, in some form, reflect the university policy under which registration schedulers must operate. Some of the secondary objectives are:

1. Section Balancing: Proponents of section balancing claim that if there are ten sections of a course then each of them should have approximately the same number of students. Opponents however feel that possibly seven or eight should be filled to capacity leaving the option to cancel the additional two or three sections. As will be seen later, both concepts achieve section balancing and with proper management little difference will exist between the two approaches.
2. Student Priorities: Sometimes it is desirable to give some students priorities over other students. Many institutions feel that seniors should have priorities over juniors and juniors over sophomores and so on.
3. Request Priorities: Within a set of student priorities, if any, it is possible to assign priorities to requests such that the second request of the first student has lower priority than the first request of the last student. More complicated schemes of similar priorities may be defined (3). An altogether different type of priority might be to give preferential treatment to majors, i.e., math majors get the first option on math courses, English majors on English courses and so on.

4. Time Compactness: In certain scheduling situations it is important to ensure that the entire student schedule is as compact as possible. This is true for schools where a large percentage of students commute to classes. It is desirable for these students to have little or no free time between classes.

SYSTEM A

In its most primitive form CAR consists of a unit record based class-card type system. Essentially, this system has a punched card associated with every seat in each section. Students wishing to enroll in various sections merely pick up the right card, add their own identification to it in the form of another punched card and turn it in for computer processing. Such a system is usually operated from an "arena" where different departments have their "stations." Students queue up at the departmental stations to pick up the punched cards associated with the sections they wish to enroll in and turn in these cards at some central location where they also pay their fees. This completes the registration process. A system like this represents the first step in the automation process and all CAR systems that essentially follow this pattern will be called System A.

Let us, very briefly, examine the characteristics of System A. The system is very simple. The administrative functions associated with running the system are easy to monitor. A system like this would work remarkably well for institutions with low enrollments. However, it has two major drawbacks:

1. There is no provision in the system to record correctly the demand for various courses. The lack of this data prevents the administration from reallocating its resources to respond to student needs for courses. The students must adjust their course selections to an a priori allocation of resources. (# sections, section sizes, etc.)
2. Adjustments to student schedules are extremely difficult to accomplish, particularly if this involves second order interactions of section changes (i.e., in order to enroll in one section, one or more of the sections currently on the schedule have to be shuffled around).

SYSTEM B

A better system (compared to System A) would be one that permits the administration to reallocate its resources to meet student demands and permits the students quickly and conveniently to readjust their schedules. As discussed earlier, there will always be certain situations where demands for courses cannot be met, or should not be met. This and the fact that not all students can get all courses at their requested times necessitate certain changes in the schedule as requested by the students. These changes are of two types: section changes and course changes. Section changes reflect a situation where the section requested by the student is full or cancelled and he is forced to enroll in an alternate section of the same course. Course changes come about if all sections of the requested course are full or represent a conflict so forcing the student to choose an alternate course. (Course changes may also result from failed courses or curriculum changes but timewise this happens later in the process). Thus a better system has three requirements:

1. Permits the university accurately and conveniently to respond to student demands for courses.
2. Allows for easy and possibly automatic section reassignment.
3. Has provisions to handle alternate course requests for courses that are full.

Any system that meets these three requirements will be called System B. Most universities use some form of System B. Registration systems that are totally on-line and do not have an effective method of recording student demands cannot be classified into System B.

CHARACTERISTICS OF SYSTEM B

In order to accomplish the first requirement it is necessary to have some sort of preregistration procedure. This then gives the university the ability to tally total demands for all courses, (or actually schedule students into classes) and determine any changes that need to be made to the Master Class Schedule. This process of modifying the Master Class Schedule to more closely meet the demand is generally accomplished through a "management report." On reviewing the management report the department heads add or delete some sections and increase or decrease the capacities of other sections. Since this process is done with full knowledge of the demand for the course, and since the objective is to more closely meet the demand, it may be verified that any logical action will have the ultimate effect of balancing sections.

The second requirement dictates the need for a registration scheduler. The registration scheduler must meet the secondary objectives defined earlier. In view of the above discussion, if a management review process exists then it is not necessary for the scheduler itself to have any section balancing capabilities for this, if required, can be forced upon it. Other details about schedulers are presented in the technical section.

The first two requirements, as seen above, spell out the need for a preregistration procedure and a registration scheduler. However, it is the third requirement of alternate course requests that accounts for the different forms that System B can take. Among these different forms are the following:

1. A preregistration procedure in which the student requests a given set of course sections. These requests are processed by a registration scheduler. Students unable to get complete

schedules handle alternate course requests through some form of arena registration.

2. A preregistration procedure in which the student not only specifies his first choice for course sections but also specifies alternate courses in case the requested ones are not available. These requests are then handled by a registration scheduler. Since students have already specified alternate course requests during preregistration, conceivably the activity in any follow up registration is minimal.

3. A procedure similar to (1) above, except that the alternate course requests are handled by a means other than "arena" registration. This generally, though not always, is some sort of on-line procedure.

The first form is a logical extension of System A. It sufrers, though to a lesser degree, from the same drawbacks as System A. Since 75% to 85% of the students have satisfactory schedules through the preregistration process, the "arena" operations are less painful. The second form tries to circumvent the problem by having all students make alternate choices along with their primary requests (only 25% need them) and uses this information to minimize and possibly eliminate the "arena" operations. The third form on the other tries to automate the arena operations. The form this automation process takes becomes a logical substitute for the arena. The success of such a substitution would depend largely on the sophistication of the underlying computer programs.

The greatest merit of the second form is that it consists entirely of batch processing. Its major drawback is that it requires all students to make alternate course choices whereas only about one-fourth need to do so. The third form uses an effective combination of both batch and on-line processing. The batch processing handles the bulk of the load and carries the system through its "planning" phases and the on-line portion ties the loose ends. Such a system, though desirable, requires both an on-line capability and twice the amount of software. In any event System B, irrespective of the form it takes, rests heavily on a Registration Scheduler.

TECHNICAL SECTION

The objective of the technical section is to shed some light on certain important concepts that are common to most registration schedulers. A reader not interested in the mechanics of scheduling operations should go directly to the next section.

Mobility Groups

The concept of mobility groups is fundamental to registration schedulers. A mobility group defines a set of course sections that belong to the same course and can act as logical substitutes for each other. It is necessary for the scheduling process to be able to identify the mobility group for each section on the master class schedule. The method of identifying the mobility group defines both the type of processing and the structure of the master class file. Mobility groups may be identified in the following four ways:

a. List Store: Associated with each section we store a list of all other sections that belong to the same mobility group. The excessive storage requirements and file maintenance due to additions and deletions should generally prohibit the use of this method.

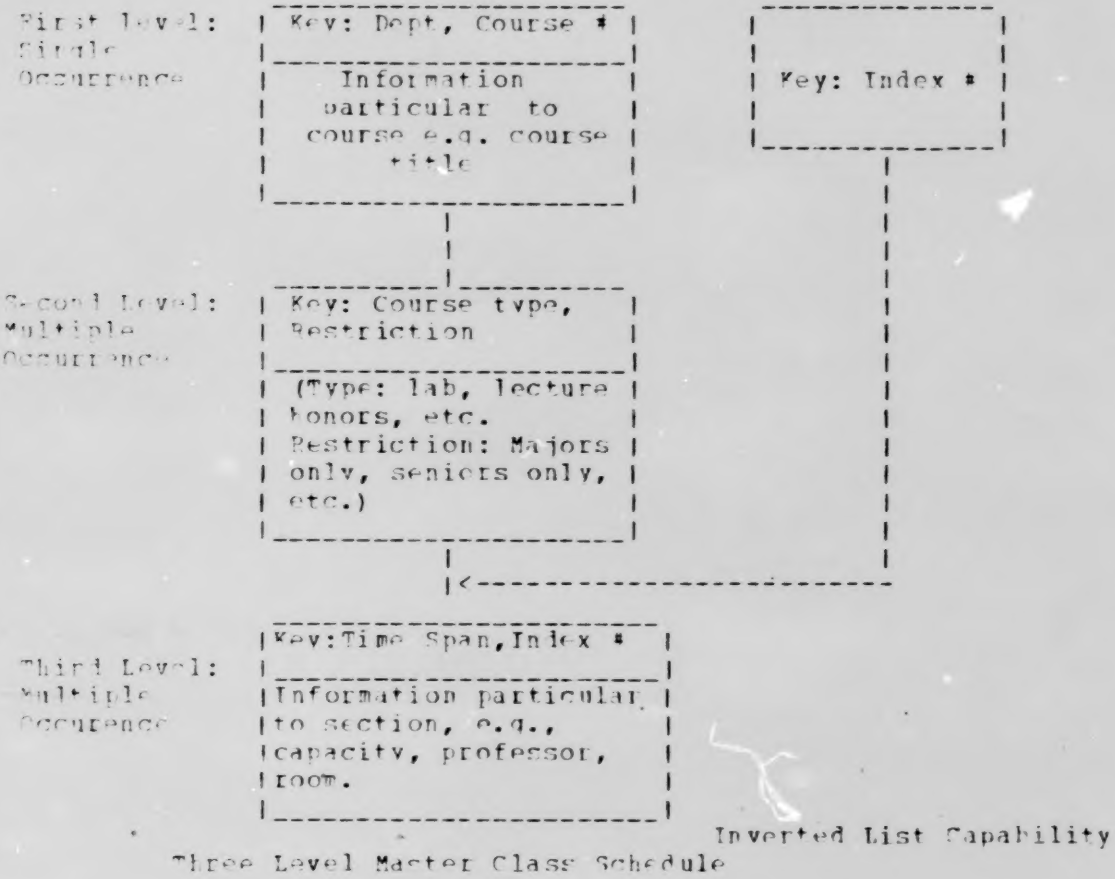
b. Chain Store: Associated with each section we store the next section that belongs to the same mobility group. The last section in the mobility group points to the first section, thus closing the chain. This procedure conserves storage space and reduces file maintenance but increases the processing necessary in the schedluer program.

c. Sequential Store with Pointers: This is currently the most popular method of identifying mobility groups. All sections belonging to the same mobility group are grouped together and each one carries a pointer to the beginning and the ending points for this group. This method is clearly better than the first two methods, but still leaves much to be desired. In such a data organization each time an addition is made to any mobility group, the pointers for all mobility groups that follow it have also to be modified. This lends a certain amount of inflexibility to the system. This inflexibility becomes particularly troublesome if in additon, as in the Utah system, the edited requests are identified by a pointer into the mobility group. A change in the master class schedule now requires not only a change in pointers in many mobility groups but also a change in the pointers for all requests. Nevertheless, so long as a registration system is limited to computer cards and tape files, this seems to be operationally the most acceptable approach.

d. Hierarchial Storage: The master class schedule may be arranged hierarchically to automatically identify the mobility groups. Such a structure would overcome the inflexibility of the previous method. A possible organization of the data is shown in Figure 1.

The data is arranged in three levels. The second and third levels have multiple occurences under a single first level segment. (An additional first level segment may be used to provide inverted list capabilities.)

The key fields in the segments determine the logical sequence of the file and play an important role in the overall scheduling process. The first and the second levels of the segments completely define the mobility group. The key fields in the third level determine the arrangement of the sections within a mobility group. Since the time span is a part of the key, it may be used directly to access or avoid sections meeting at particular time spans. The ability to do this considerably reduces the effort required to generate a conflict free schedule.


First level: Single Occurrence
Key: Dept, Course #
Information particular to course e.g. course title
Key: Index #
Second Level: Multiple Occurrence
Key: Course type, Restriction
(Type: lab, lecture honors, etc. Restriction: Majors only, seniors only, etc.)
Third Level: Multiple Occurence
Key:Time Span,Index #
Information particular to section, e.g., capacity, professor, room.
Inverted List Capability
Three Level Master Class Schedule

Hierarchial File Layout
Figure 1

Time Strings

There are two methods of storing times at which classes meet.

a. Bit String: The time at which a section meets can be represented by a bit string of appropriate length, each bit representing a fifteen minute time interval (7) (10). The bit string can be anywhere from 30 to 60 bytes long. If there are as many as 3000 sections being offered, the storage requirements become appreciable. Since the number of unique bit strings in the set of 3000 will be considerably smaller than 3000, it is advisable that only unique bit strings be stored and the time field simply point at the appropriate bit string. This will reduce the storage requirement.
   The use of bit strings facilitates the determination of conflicts in a student's schedule. Only a single compare is needed to determine the conflict. However, since most higher level languages do not permit bit manipulations, the programming of this method is restricted to the use of assembler language.

b. Day Time Codes: If it is necessary to do all the programming in COBOL then the bit string method of data storage cannot be used. In such cases it is possible to code both the course meeting time and days. (See Utah System.) Generally it is sufficient to use a single alphanumeric character to code all begin times. The single position allows for 40 possible begin times. The end times of the course may be similarly coded. Coding the days is a little more complicated. If classes meet Mondays through Fridays, then there are five possible days giving 32 possible combinations of days. Thus a number between 1 and 32 may be used to code the days. (Six day weeks require 1 through 64.)

   Using this coding scheme, in order to determine a conflict we first check to see if the days conflict. Having generated the codes, the easiest way to do this is by using a conflict table (see Utah System). The conflict table is a 32 x 32 table of blanks and "X's". A X in any positon marks a conflict. If a day conflict exists, the time conflict can be determined by comparing the begin and end times of the two courses. This method of coding requires less storage but more processing to determine conflicts.

Algorithms

There are many scheduling algorithms in existence (7), (9), (11), (12). A copy of the last two algorithms is included in APPENDIX B. Scheduling algorithms can be placed into two categories; "combinatorial search" algorithms and "least resistance path" algorithms. Combinatorial search algorithms essentially look at every possible combination of course sections to see if a conflict free schedule is available. The computations involved may be reduced by properly arranging the search pattern. A detailed presentation of a version of this algorithm may be found in (7).

The algorithms that use the least resistance path do not check for all possible combinations. The usual "backtracking" procedure common to combinatorial algorithms is not required for these algorithms. The essential steps in this type of algorithm are:

a. Arrange all course requests in the ascending order of number of non-repetitive open sections (open sections that meet at different times).
b. If the minimum number of open sections is zero, the course cannot be assigned. If the number is greater than zero, choose a section that has a minimum number of conflicts with sections of other courses. This defines the least resistance path. Assign the chosen section.
c. Remove from the unassigned courses all sections that conflict with the one just assigned. Recompute the minimum number of open sections and return to step b.

Within this framework it is possible to introduce many subtleties that in some form meet the secondary objectives defined earlier. The method of selecting non-repetitive open sectons in step 'a' can introduce 'section-balancing' capabilities into the system. Applying the algorithm in parts can introduce a "priority" for students original section requests (11). Appropriate arrangement of sections in step 'a' and alternate selection scheme in step 'b' can introduce "time-compactness" property into the schedules.

SYSTEM C

Most universities are at a level of System B. Improvements in System B will tend to be more technological than conceptual. The administration of many versions of System B generally requires a continuous fire fighting effort. Many systems seem to be an endless string of prints, sorts and merges. The effort required to insure input data integrity is inordinately high and the setbacks due to incorrect data are higher. Moreover, there is now an increasing sensitivity towards data security and confidentiality of information. This requires a change in the operating procedures of many registration systems. Improvements therefore should be such as to require a minimum of human attention to run the system. This would require the introduction of the following concepts:

1. A monitor for the registration system. A program that runs other programs.
2. A policy file that defines all default situations for processing. This should also include actions to be taken in case of time delays.
3. The ability to automatically assign inter-relationships between data elements. To a great degree this can be achieved by proper file structure.
4. The ability to independently update any file at any time without compromising the system. This may be achieved either by

having all files on-line or by queuing update transactions and later processing these through a batch program.

5. Once the master class schedule is finalized, to permit the students to alter their schedules without any constraints, so long as these changes are done before specified deadlines.

In order to accomplish some of these objectives it becomes necessary that the registration system be a part of a comprehensive information management system for the university. For a large university such a step is necessary and possibly inevitable.

References

1. V. A. Abell, "Perdue Academic Student Scheduling," 1965. (Perdue University, Lafayette, INdiana)

2. "The Computer Assisted Registration System for California State Universities and Colleges", 1973. (Prepared by Computer Center, California State University, Sacramento)

3. "Kent State University Computerized Registration System", 1972 (Prepared by Management Information Systems, Kent State University, Kent, Ohio)

4. "OASIS System Description", 1970 (Prepared by Project INFO, Encinia Hall, Stanford University, Stanford, California)

5. "Computer Aided Registration System", 1971 (Prepared by Office of Registrar, University of Houston)

6. R. M. Hinton and K. E. Fuhrup, "On-line Advanced Registration" (St. Petersburg Junior College) Proceedings of 18th Annual CUMREC Conference 1973

7. [illegible]. I. Merher and M. D. Zisman, "Responsive Registration and Scheduling at University of Pennsylvania," Proceedings of 18th Annual CUMREC Conference, 1973.

8. [illegible]. Jenkins, "Continuous Advisement and Registration" (Central State University, Oklahoma City) Proceedings of 18th Annual Cumrec Conference 1973.

9. W. K. Winters, "A Scheduling Algorithm for a Computer Assisted Registration System", 1971, Communications of the ACM, Vol. 14, Number 3, March 1971.

10. "OSCAR: Student Scheduling System", 1972 (Prepared by Co-ordinator Systems Development, University of Massachusetts)

11. V. Chachra, "Documentation of Registration System at Virginia Tech", 1973 (VPI & SU, Blacksburg, Va)

12. V. Chachra, "Student Registration Scheduling Program", (Utah Scheduling Algorithm), Virginia Tech Computing Center Technical Paper #VT 71152, 1971.

APPENDIX A

Table 1 - List of Colleges
Table 2 - Registration Data
Table 3 - ID Card Information
Table 4 - Terminal Operations
Table 5 - Hardware

TABLE - 1

LIST OF COLLEGES AND UNIVERSITIES THAT RESPONDED TO THE QUESTIONAIRE

1. Auburn University
2. Central Piedmont Community College
3. Houston Community College
4. Indiana University
5. Kent State University
6. L. S. U., Baton Rouge
7. L. S. U., New Orleans
8. Montclair State
9. North Illinois University
10. Ohio State University
11. South Illinois University
12. Stanford University
13. S. U. N. Y. at Buffalo
14. Syracuse Univeristy
15. University of Florida
16. University of Georgia
17. University of Houston
18. University of Illinois
19. University of Kentucky
20. University of Massachusetts
21. University of Michigan
22. University of Minnesota
23. University of Nebraska
24. University of New Hampshire
25. University of North Carolina
26. University of Oklahoma
27. University of Oregon
28. University of South Carolina
29. University of Tennessee
30. University of Texas
31. University of Utah
32. University of Wisconsin
33. Western Kentucky University
34. West Virginia University
35. University of Virginia
36. Virginia Polytechnic Institute and State University

TABLE - 2

REGISTRATION DATA

Question #	1	2	3	3	4	5	5	6	7	8	8	8	8	9	9	9
			CAR							Requests				CAR System		
School	Population (K)	# Campuses	CAR	Years in Use	Developed	Quarter	Semester	Pre-registration	Section Requests	Punch Cards	Optical Scan.	Terminals	Other	Off-line	On-line	Combination
Auburn	15.0	2	Y	5	Y	Y		Y	Y	Y				Y		
Cen. Pied.	15.0	1	Y	2		Y			Y	Y				Y		
Indiana U.	69.0	9	Y	5	Y		Y	Y	Y	Y				Y		
Kent State	19.7	6	Y	2	Y	Y		Y	Y		Y			Y		
L.S.U., B.R.	22.0	1	Y	3	Y		Y	Y	Y	Y				Y		
Montclair	13.5	1	Y	4			Y	Y	Y	Y				Y		
N. Illinois	22.0	1	Y	2			Y	Y	Y	Y				Y		
Ohio State	50.0	6	Y		Y	Y		Y	Y	Y						Y
S. Illinois	19.0	2	Y	7	Y	Y		Y	Y	Y				Y		
Stanford	11.6	6	Y	2	Y	Y		Y	Y	Y						Y
S.U.N.Y.	24.0	4	Y	3	Y		Y	Y	Y		Y			Y		
Syracuse	19.0	5	Y	6	Y		Y		Y	Y				Y		
U. of Fla.	23.5	1	Y	5	Y	Y		Y	Y			Y			Y	
U. of Ga.	20.0	1	Y	1		Y		Y	Y			Y				Y
U. of Houston	26.4	1	Y	2	Y		Y	Y	Y		Y			Y		
U. of Ill.	32.4	4	Y	8	Y		Y	Y	Y	Y				Y		
U. of Ken.	19.6	15	Y	5	Y		Y	Y	Y	Y				Y		
U. of Mass.	28.6	4	Y	2	Y		Y	Y	Y		Y					Y
U. of Mich.	41.2	3	Y	2	Y		Y	Y	Y		Y					Y
U. of Minn.	50.0	6	Y	11	Y	Y			Y	Y						Y
U. of Neb.	21.0	3	Y	4	Y		Y	Y	Y		Y			Y		
U. of N. H.	9.4	1	Y	8	Y		Y	Y	Y	Y				Y		
U. of N.C.	19.2	2	Y	5	Y		Y	Y	Y	Y				Y		
U. of Okla.	19.4	2	Y	5	Y		Y	Y	Y	Y				Y		
U. of Ore.	15.4	1				Y		Y			Y			Y		
U. of S. C.	20.0	4	Y	5	Y		Y	Y	Y	Y				Y		
U. of Tenn.	25.5	1	Y	5	Y	Y		Y	Y		Y					Y
U. of Texas	39.9	1	Y	3	Y		Y	Y	Y				Y			Y
U. of Utah	24.0	1	Y	7	Y	Y		Y	Y	Y				Y		
U. of Wis.	34.9	1	Y	4	Y		Y		Y	Y				Y		
Wes. Ken. U.	11.5	1	Y		Y		Y	Y	Y	Y		Y				Y
W. V. U.	17.0	3	Y				Y	Y	Y				Y			
U. of Va.	13.5	2	Y	5	Y		Y	Y	Y	Y				Y		
V.P.I.	15.5	1	Y	1	Y	Y		Y	Y		Y	Y				Y

TABLE - 2 (con't)

REGISTRATION DATA

Question #	10	11	12	13	14	15	16	16	16	16	18	18	18	18	18	19	20
							Correction Changes				CAR Integrated With						
School	Editing	Conflict Res.	% Satisfactory	Mgmt. Info.	Mail Class Tkts.	Make Span (wks)	Manual	Terminal	Off-line	Other	Admissions	Student Accounts	Grade Reports	Graduation Analysis	HEW Reports	Classroom Assign.	Space Alloc.
Auburn	Y	Y	85	Y	Y	4	Y				Y	Y	Y	Y	Y		
Cen. Pied.	Y		90	Y	Y	2			Y			Y	Y		Y		
Indiana			70	Y			Y				Y	Y	Y	Y	Y		
Kent State	Y		75		Y	2	Y				Y	Y	Y	Y	Y		
L.S.U., B.R.	Y	Y	92	Y						Y	Y	Y	Y		Y		
Montclair	Y	Y	60		Y	10	Y				Y	Y	Y		Y		
N. Illinois	Y	Y	75	Y	Y	5					Y	Y	Y				
Ohio State	Y	Y	85		Y	2			Y		Y	Y	Y				
S. Illinois			99	Y	Y	1				Y	Y	Y	Y	Y	Y	Y	Y
Stanford	Y		99	Y	Y	4	Y				Y	Y	Y	Y	Y		
S.U.N.Y.	Y	Y	85	Y	Y	3				Y	Y	Y	Y	Y	Y		Y
Syracuse			90	Y						Y	Y	Y	Y	Y	Y		
U. of Fla.	Y		99	Y	Y	2	Y										
U. of Ga	Y		80					Y			Y	Y	Y	Y			
U. of Houston	Y	Y	80	Y	Y	1	Y				Y	Y	Y		Y		
U. of Ill.	Y	Y	70	Y	Y	3			Y		Y	Y	Y	Y	Y	Y	Y
U. of Ken.	Y	Y	85	Y	Y	3				Y	Y	Y	Y				
U. of Mass.	Y	Y		Y	Y	11	Y				Y	Y	Y		Y		
U. of Mich.	Y			Y	Y	8			Y			Y	Y		Y		
U. of Minn.				Y						Y	Y		Y		Y		
U. of Neb.	Y	Y	85	Y	Y	3	Y	Y			Y	Y	Y		Y		
U. of N. H.	Y	Y	87	Y	Y				Y								
U. of N. C.	Y		98	Y			Y				Y	Y	Y	Y	Y		
U. of Okla.	Y		75	Y	Y	2	Y				Y	Y	Y	Y	Y		
U. of Ore.											Y	Y	Y	Y	Y		
U. of S. C.	Y			Y			Y				Y		Y	Y	Y		
U. of Tenn.	Y	Y	80	Y		5		Y			Y	Y			Y		
U. of Texas	Y	Y	98	Y	Y	3				Y	Y	Y	Y	Y	Y		
U. of Utah	Y	Y		Y	Y	4	Y				Y		Y	Y	Y		
U. of Wis.	Y		92	Y			Y				Y	Y	Y	Y	Y	Y	Y
Wes. Ken. U.	Y		99	Y	Y					Y	Y	Y	Y		Y		
W. V. U.																	
U. of Va.	Y		45	Y		4				Y	Y	Y	Y	Y	Y		Y
V. P. I.	Y	Y	75	Y	Y	3		Y			Y	Y	Y	Y	Y		

TABLE - 3

ID CARD INFORMATION

School	Picture	Machine Readable	Issued Quarter	Semester	Yearly	Once	Revalidated Punch	Embossed	Magnetic	Gummed Labels	Uses Dining Hall	Library	Athletic	Cultural	Infirmary
Auburn	Y				Y					Y	Y	Y	Y	Y	
Cen.Pied.			Y							Y		Y		Y	
Indiana U.															
Kent St.	Y	Y				Y		Y			Y	Y	Y	Y	
L.S.U., B.R.	Y	Y			Y					Y	Y	Y		Y	Y
Montclair	Y				Y					Y			Y	Y	
N. Illinois	Y					Y				Y	Y	Y	Y	Y	
Ohio St.	Y					Y				Y	Y	Y	Y	Y	Y
S. Illinois	Y	Y				Y						Y			
Stanford	Y					Y				Y	Y	Y	Y	Y	
S.U.N.Y.					Y						Y	Y	Y	Y	
Syracuse	Y	Y				Y		Y			Y	Y	Y		
U. of Fla.	Y					Y				Y		Y	Y	Y	Y
U. of Ga.	Y	Y				Y						Y	Y	Y	Y
U. of Houston	Y					Y		Y			Y	Y	Y	Y	
U. of Ill.	Y	Y			Y	Y		Y		Y		Y	Y		Y
U. of Ken.	Y				Y		Y				Y	Y	Y	Y	Y
U. of Mass.	Y	Y				Y				Y		Y		Y	
U. of Mich.		Y				Y		Y				Y	Y	Y	
U. of Minn.	Y					Y				Y		Y	Y	Y	Y
U. of Neb.						Y				Y		Y	Y	Y	
U. of N.H.	Y					Y				Y	Y	Y	Y		
U. of N.C.	Y	Y				Y				Y		Y	Y	Y	
U. of Okla.	Y					Y		Y				Y	Y	Y	Y
U. of Ore.	Y	Y				Y						Y			
U. of S.C.	Y			Y									Y	Y	
U. of Tenn.	Y	Y				Y		Y			Y	Y	Y	Y	
U. of Texas	Y	Y			Y		Y	Y			Y	Y	Y	Y	
U. of Utah	Y	Y				Y				Y	Y	Y	Y	Y	
U. of Wis.	Y	Y				Y		Y				Y	Y		
W. Ken. U.	Y	Y				Y		Y				Y	Y	Y	Y
W.V.U.	Y				Y					Y		Y	Y	Y	Y
U. of Va.	Y					Y				Y		Y		Y	
V.P.I.	Y				Y		Y	Y			Y	Y	Y	Y	Y

TABLE 4 - TERMINAL OPERATIONS

Ques. #	Question	Stanford U.	U. of Florida	U. of Georgia	U. of Minn.	U. of Nebraska	U. of Tenn.	U. of Texas	Western Ky. U.	V.P.I. & S.U.
28	Conversational	Y				Y	Y	Y	Y	Y
30	Time Conflicts		Y	Y			Y			Y
32	Processing time (min)		.3	1.0		1.2	1.8		2.0	2.0
34	# Students/ day		2500	1100		340	7000		600	800
35	# Terminals		4	6	4	8	24		6	10
36	# Days Used		9	10	30	5	7		5	7
38	Central Loc.		Y			Y	Y		Y	
39	Hard Copy Capability	Y								
39	Video	Y		Y	Y	Y		Y	Y	Y
40	Cobol	Y						Y		Y
40	Fortran									
40	PL/1								Y	
40	Assembler	Y	Y	Y	Y	Y	Y		Y	
41	Core (K)	6	40	80	100	62	64	200	40	16
42	MIS	Oasis						Total		IMS

Notes: a. Question number corresponds to the questionaire in Appendix B.
b. All terminals were also used for purposes other than registration.
c. In all cases terminals were not operated by students.

TABLE - 5

HARDWARE

School	CPU	Operating System	Core (K)	Terminals	Tape Drives	Disk Drives	Printer	Reader/Punches	Optical Scanning
Auburn	IBM 360/50		512						
Cent. Pied.	IBM 1401		16			2	1	1	1
Houston C.C.	IBM 360			6	5	8	2	1	1
Indiana U.	IBM 360/120 IBM 370/155	HASP/MVT	1024	15	8	8	2		
Kent St.	Honeywell 1250		131		6	4	2	2	
L.S.U., B.R.	IBM 360/40	DOS	256	20					
L.S.U., N.O.	IBM 1401		8						
Montclair	IBM 360/67	OS	1500		10	3			
N. Illinois	IBM 360/67	MVT/HASP	512						2
S. Illinois	IBM 370/155								
Stanford U.	IBM 370/145	DOS				8			
S.U.N.Y.	UNIVAC 1106	EXEC8			4	8			
Syracuse U.	IBM 370/158	MFT	1024		2	16	2		
U. of Fla.	IBM 370/165			125		16			
U. of Ga.	IBM 360/65	HASP	3000						2
U. of Houston	UNIVAC 1108		64		6		2	2	
U. of Ill.	IBM 370/155	MVT/HASP	1500	2	22	16	3	2	
U. of Ken.	IBM 360/65	MVT/HASP	768	10	5	2	2	2	
U. of Mass.	IBM 360/145 IBM 360/30	DOS							
U. of Mich.	IBM 370/155	MFT/HASP	1000	16	8	12			
U. of Minn.	IBM 360/50	OS	384			2			
U. of Neb.	IBM 360/65	HASP			8	24			
U. of N.H.	IBM 360/50	MVT/HASP	156	23	5	8		1	
U. of N.C.	UNIVAC 70/7		512		6	12	2	2	1
U. of Okla.	IBM 360/50	HASP	512	9					
U. of Oregon	IBM 360/50	MFT/HASP	256		5	3			
U. of S. C.	IBM 370/155	DOS	1500						
U. of Tenn.	IBM 360/65	HASP	750	1	4	28	2		
U. of Texas	IBM 370/155	MVT			7	8			
U. of Utah	IBM 360/40	DOS	384	18	3	12	1		
U. of Wis.	IBM 370/155	OS	1500						
W. Ken. U.	IBM 360/40	DOS		29	3	9	1	1	
W. V. U.	IBM 360/75	MVT							
U. of Va.	IBM 370/145	MFT/HASP	384	1					
V.P.I.	IBM 370/158	MVT	2000 1500	65	5	16	3	3	2

. THE UTAH ALGORITHM

Define the following variables:

1. $C_i\{c_{ij}, j=1,\text{-----},s\}$ a set of sections of course i.
2. $R\{r_i, i=1,\text{-----},q\}$ a set of student requests where $r_e \in C_i$ and $r_e = c_{ij}$ for some i, j. If a requested section is cancelled then $r_e \in C_i$ but $r_e \neq c_{ij}$ for any j.
3. n_i is the remaining capacity of course C_i; n_{ij} is the remaining capacity of section c_{ij}.
4. T_{ij} is the set of blocks in the weekly calendar when section c_{ij} meets.
5. F_{ij} is the set of sections conflicting with section c_{ij} and is defined in step 10 of algorithm.
6. O_i is the set of open sections associated with the course to which the request r_i belongs.
7. N_i is the number of elements in O_i.
8. Y_i is a subset of O_i consisting of sections which cannot be used because they conflict with courses already assigned.
9. a_i contains the assignment made for request r_i.

   If $a_i = \begin{cases} c_{..}, \text{ means request } r_i \text{ has been met by } c_{..} \\ 0 \quad \text{implies an overflow} \\ \text{negative implies a conflict} \end{cases}$
10. $w_i = \begin{cases} 0 \text{ if request } r_i \text{ is not a linked course} \\ 1 \text{ if request } r_i \text{ is a linked course} \end{cases}$
11. R' is a subset of R consisting of sections which are closed.
12. $\sim$ is a symbol for linked courses. Thus $C_i \sim C_j$ means that both C_i and C_j must be scheduled together.

Algorithm

```
A     Initialize variables.  Read student requests into set R.
1.    Delete duplicate course requests:
      IF $\exists$ e, k, $e \neq k$ such that
            $r_e \varepsilon C_l$, $r_k \varepsilon C_l$ for some l,
      THEN delete $r_e$ from set R.
2.    Check if both linked courses are requested, if not add one:
      IF $\nexists$ k $\ni r_k \varepsilon R$ and $r_e \varepsilon C_l$, $r_k \varepsilon C_j$ such that $C_l \sim C_j$
        THEN add $r_k$ to set R.
                              Do this for each $r_e \varepsilon R$ such that $r_e.W_e \neq 0$
                              Go to 8.
        ELSE go to 3.
3.    IF $\exists$ $r_e \varepsilon R$ such that $r_e \varepsilon C_l$, $r_e \neq c_{lj}$  $\forall_j$
            GO to 8.
      ELSE go to 4.
4.    IF $n_{lj} > 0$  $\forall r_e \varepsilon R$, $r_e = c_{lj}$
        THEN  $a_e = r_e$  $\forall e$. Go to 7
      ELSE go to 5.
5.    Define $R' \varepsilon R$;  $R' \{ r_e / r_e = c_{lj}, n_{lj} = 0 \}$
6.    IF for each $r_e \varepsilon R'$ $\exists$ $c_{lk}$ $\ni$ $\{ r_e = c_{lj}, n_{lk} > 0, T_{lk} = T_{lj} \}$
        THEN define
            for each $r_e \varepsilon R'$,      $a_e = c_{lk}$
            for each $r_e \notin R'$,      $a_e = r_e$,  go to 7
      ELSE go to 8.
```

```
7.  For each $a_e=c_{ij}\neq 0$     define $n_{ij}=n_{ij}-1$   [REGISTERED]
    For each $a_e=0$                RECORD OVERFLOW
    For each $a_e<0$                RECORD CONFLICT
    Return to A
8.  For each $r_e \varepsilon R$ define set $O_e$ $\{j : r_e=c_{mn}, c_{mj} \varepsilon C_m, n_{mj}>0,$
    IF $k \varepsilon O_e, T_{mj}\neq T_{mk}\}$
9.  Determine $N_e=|O_e|$
10. Determine conflicts:
    for each $r_e \varepsilon R$, $j \varepsilon O_e$ determine
      $F_{ej}\ \{(k,m) \mid k\neq e, r_k \varepsilon R, m \varepsilon O_k, T_{mk} \cap T_{ej}\neq 0\}$
11. Determine k such that $N_k=\min_i N_i$,  $r_k \varepsilon R$
    Remove $r_k$ from set R.
    IF $N_k=0$ go to 14
    ELSE go to 12
12. Determine section with minimum conflicts:
    Find j such that $j \varepsilon O_k$, $j \not\varepsilon Y_k$,  $|F_{kj}|=\min|F_{ki}|$
    IF no such j exists, go to 15
    ELSE $a_k=c_{kj}$.  Go to 13
13. Update conflicts & return:  IF R=0  go to 17
    ELSE for each $r_e \varepsilon R$ define $Y_e\{m \mid (e,m) \varepsilon F_{kj}\}$
    Remove from all $F_e$. elements (k,·).  Go to 11
14. No open section available:  $a_k=0$  go to 16
15. Conflict:  $a_k$=negative.  Go to 16.
16. IF R=0, go to 17.  ELSE from all sets $F_e$. such that $r_e \varepsilon R$
    remove elements (k,.).  Go to 11
17. Check for assignment of linked courses; for each pair
    $r_e$, $r_k \varepsilon R$ such that $\{r_e \varepsilon C_i, r_k \varepsilon C_j, C_i \sim C_j\}$
    IF $a_k$ or $a_e\neq 0$  THEN $a_k$ and $a_e=\min(a_k,a_e)$  Go to 7.
```

THE VIRGINIA TECH ALGORITHM

Program Description/Logic

Program Name: Batch Scheduler Program #: STRR150

The scheduler program reads the edited requests and processes these requests against the timetable. If there is room in every requested section, the student is directly scheduled into these sections. Thus if the requested set of course sections have a time conflict, the student will receive a schedule with a time conflict. It is therefore the student's responsibility initially to request conflict free schedules.

In case one or more requested sections are full or cancelled, then the program first tries to allocate the student to an open section of the requested course that meets at the same time. If all the student requests cannot be met by the above process, then in order to give the student a complete schedule it is necessary to re-schedule those courses requested by him. This later scheduling process (resolving conflicts) forms the major part of the schedule program.

The mechanism for resolving conflicts has one rather simple exception. If the student has only one request then, allocating his request to any available time slot (i.e., any open section) will suffice as there cannot possibly be a conflict. Barring this one exception, the program has to go through rather complicated processing to resolve the conflicts. The principle underlying this processing is as follows:

a. The primary effort is to give the student the maximum number of courses.

b. In giving the maximum number of courses effort is made to give the student as many of his original requests as is possible.

c. In allocating an alternate section (section other than the requested one) effort is made to give the student the requested time slot.

d. Within a given time slot the student is assigned to the section having the maximum number of seats available. To this extent the program has section balancing capabilities. Since this is in the fourth level of pirority its effect is apt to be minimal.

The logic that accomplishes this is best understood by classifying the student's requests into three groups. Initially the first group (G1) consists of those requested sections that either

a. Have the requested section as the only open section associated with it, or

b. have no open sections.

The courses that have no open sections associated with them cannot be allocated to the student and those that have a single open section must be allocated to the student.

The second group (G2) initially consists of those requests that have capacity and are associated with more than one open section. This table is headed by the requested section followed by other open sections that meet at different time slots. In case there is more than one open section meeting at the same time, the one with the maximum capacity is selected. This is done for all requests belonging in this category.

The third group (G3) consists of those requests that have no capacity (full section or cancelled section) for the requested section but have at least one open section meeting at a time other than the requested time.

Three types of assignments are made for each section. An assignment of '1' indicates that the student is enrolled, '2' indicates that the request constitutes an overflow and '3' says that the request could not be met due to a conflict. An assignment of '0' indicates that an assignment is yet to be made.

Using the above definition, we are now in a position to discuss the scheduling algorithm. There are two phases to this algorithm. The second phase is used only if the first cannot completely schedule the student.

Scheduling Algorithm Phase I

a. In G1 the open sections are assigned '1' and the closed sections assigned '2'.

b. The requested section (the first entry) corresponding to each course in G2 is '1' and the remaining sections are assigned '0'.

c. If any of the sections assigned '1' so far are mutually conflicting, then this phase is terminated and we proceed directly to phase II.

d. All sections in G3 that conflict with any section assigned '1' in G1 and G2 are removed from G3.

e. A course with a minimum number of open sections in G3 is selected. If there is more than one such section, the first one is selected.

f. Three cases need be considered. First, if the minimum in step 'e' is zero then the request is assigned '3' and moved to G1. At this point processing of phase I is terminated and we go directly to phase II. Second, if the minimum is one, then it is assigned '1' and moved to G1. All sections in G3 conflicting with this assignment are removed from the table. Third, if the number of open sections if greater than one, then the section with the least number of conflicts with other courses in G3 is selected. This selected section is assigned '1' and moved to table G1. All other sections of this course and sections of other courses conflicting with this are again removed from table G3.

g. Steps e and f are repeated till table G3 is empty.

h. All assignments of '1' in table G2 are moved to table G1. At this point the student has a complete schedule and phase II is not necessary. Scheduling with phase I insures that the student gets as many of his originally requested sections as possible.

Scheduling Algorithm Phase II

Phase II is necessary if phase I cannot provide a complete schedule for the student. The steps in phase II are similar to those in phase I except that here we work with both tables G2 and G3. The details are as follows.

a. In table G1 the open sections are assigned '1' and the closed sections assigned '2'.

b. All entries in G2 and G3 are assigned '0'.

c. For each assignment of '1' in G1 all those entries in G2 and G3 that conflict with it are removed from the two tables.

d. A course with a minimum number of open sections in G2 or G3 is selected. If there is a tie preference is given to the contents of table G2 over those in table G3.

e. Three cases need be considered. First, if the minimum in step 'd' is zero then the request is assigned '3' and moved from its present position to table G1. Second, if the minimum is one then the request is assigned '1' and moved to table G1. Also, all sections in G2 and G3 that conflict with this assigned section are removed from the table. Third, if the minimum is more than one, then for each of these, the number of conflicts with other courses in G2 and G3 is computed. The section having the least number of conflicts is selected. This section is assigned '1' and moved to table G1 and the other sections of this course are ignored. Again, all sections contained in G2 and G3 that conflict with the section moved to G1 are removed from the table. Thus for each assignment, the contents in table G2 and G3 are reduced.

f. Steps 'd' and 'e' are repeated till both tables G2 and G3 are empty.

g. The contents of table G1 give the student's final schedule.

SYSTEM 2000 AS A DBMS FOR COLLEGES AND UNIVERSITIES

Charles J. Lewis
Data Management Analyst
Technical Marketing Services
MRI Systems Corporation
Austin, Texas 78766

"SYSTEM 2000 AS A DBMS FOR COLLEGES AND UNIVERSITIES"

I. Data Base Management vs. File Management and Traditional Data Processing.

A. File Management and Traditional Processing

File management systems are record oriented. The user accesses the entire record regardless of the number of data items in which he is interested. Updating of data fields causes record reformatting and often physical repositioning of the entire record. Regardless of the quantity of information desired, the entire file is usually passed.

File management and traditional data processing systems are program oriented. The storage of information is directly tied to the output of one program, and dictates the input of another. If a data field is added, deleted, or changed in a file, all programs accessing that file must be altered.

Programs require development times during which the end user is "locked out" of the application system design.

A single user environment is typical of file management and traditional data processing. Concurrent update is impossible for most data structures, as is sharing of object code.

B. Data Base Management

Data Base Management Systems are data oriented, instead of record oriented. Data is stored by groups, i.e., for a tree structure, data is stored for each occurence of a group of data in the tree. Storage is taken only for existing occurences of data. No padding of records is required, and additional occurences of data are easily added to any portion of the tree. Updates effect only that group of data which is involved, not the entire record.

To do this, Data Base Management Systems (DBMS) usually employ their own techniques of storage and access algorithms. In SYSTEM 2000, for example, information is stored in a group of random access files, inter-related via pointers which the DBMS, not the programmer, maintains. Multiple indexes are allowed, giving the user a variety of ways in which to qualify data. This is a necessary capability in a complete data base environment, where many users need to access the data base, qualifying on keys which are meaningful to them (i.e. all students who attend Management 356; all students whose GPA is over 3.0, etc.) Only the data in which the user is interested is processed for any retrieval or update request. This eliminates the costs of passing the entire file for any operation, and hence increases throughput.

DBMS are data base oriented instead of program oriented. Instead of a program producing a data file, the data base is created first, and then the program and natural language (English-like query and update commands) users access the data base.

If a data field is added, deleted, or changed in a SYSTEM 2000 data base definition, only those programs which work with that particular data field are affected. The alteration is transparent to the other users of the data base.

Data is stored and maintained by a DBMS so that many users can concurrently make use of the data base, either for similar or dissimilar reasons. One user may access one subsection of the data base while other users are accessing other subsections of the data base. Only the data in which a user is interested is returned to him, as opposed to the entire record.

A DBMS such as SYSTEM 2000 assumes the responsibility for handling concurrent updates, so that many users may update and retrieve from the data base (or data bases) concurrently without a loss of data integrity. Futhermore, all users may share one copy of the data management code. This reduces overall core requirements and allows more users to be serviced on a given hardware configuration.

In addition to managing data, each of the more powerful DBMS on the market today offer some or all of the following features:

- Security System
- English-like query, update, and reporting language
- Report Writer
- Backup and Recovery Procedures
- Teleprocessing Interface

These extended features offer more total capability than just an efficient and easy to use storage structure. They offer a variety of ways to access and protect data, which leads to faster implementation of application systems and more end-user involvement during the application system development phase.

II. SYSTEM 2000 Slide Presentation

SYSTEM 2000

Generalized Data Management System

Slide 1

SYSTEM 2000 is developed and marketed by MRI Systems Corp. with offices in Austin, Texas, New York City, Washington D.C., Atlanta, Detroit, Los Angeles, San Francisco, and Chicago. MRI began in 1965. The first copy of SYSTEM 2000 was installed on CDC in 1970; and the 2nd copy, revised, was installed at Sperry Rand in 1971.

Slide 2

SYSTEM 2000 is generalized:

FOR MULTIPLE SYSTEM USERS

- Executives
- Application Customers
- Computer System Designers
- Project Managers
- Programmers
- Data Entry Clerks

FOR MULTIPLE APPLICATIONS

- Accounting Systems
- Project Control
- Personnel Management
- Management Information Systems
- Manufacturing

FOR MULTIPLE HARDWARE SYSTEMS

- IBM 360/40 & 370/145
- CDC 60 & CYBER 70
- UNIVAC 1100

FOR MULTIPLE DATA ACCESS METHODS

- Ad-Hoc Query
- Standard Reports
- Application Programs
- Volume Batch Updating

FOR MULTIPLE OPERATING SYSTEMS

- OS/MFT
- OS/MVT
- VS1
- VS2
- EXEC-8
- KRONOS
- SCOPE

FOR MULTIPLE TELEPROCESSING SYSTEMS (IBM)

- Multi-faster
- CICS
- TSO
- ALPHA
- BEST
- TCAM

Slide 3

SYSTEM 2000 capabilities can be diagrammed in three modules:

I. Data Base Definition and Creation

II. Communications

   a. Immediate Access Language
   b. Queue Access Language
   c. Procedural Language Interface

III. Report Writer

Slide 4

SYSTEM 2000 is an hierarchical-tree, inverted-file Data Management System.

Defining data in its hierarchical tree relationships allows varying occurences of repeating data. For example, a record of an insurance policy might contain multiple benefits, multiple beneficiaries, repeated due dates, and multiple pay out amounts. The user is not required to specify an average number of occurrences or a maximum. Disk space is taken only as data occurs.

Storing data in an inverted file allows indexing on any field. The user may choose to select data based upon any boolean combination of values. Qualification of data which satisfies the conditions specified by the user request is accomplished in the inverted files. The qualification process produces direct BDAM pointers to the data file. No chaining of records in the data file is ever needed.

SYSTEM 2000 allows data type specification and error editing. The user may specify each field to be:

NAME or TEXT - alphanumeric
INTEGER - numeric
DECIMAL or MONEY - numeric with decimal point
DATE - month, day, year combination

SYSTEM 2000 allows any NAME or TEXT typed fields to contain up to 250 characters. The user is asked to provide a 'mode' picture size. Fields which overflow the picture size will be handled automatically.

Slide 5

SYSTEM 2000 allows two means of loading an initial data base.

1. The user may specify the commands, "DATA FILE IS INPUT: LOAD:". SYSTEM 2000 will scan the file called 'INPUT' and load the data base including construction of all inverted pointers. The input file must contain data in 'loader-string' format; that is, each value must be identified by the identifier of the field where it is to be stored.

2. Any COBOL program which currently accesses the data file can be used to load a SYSTEM 2000 data base. The program can be modified to issue a SYSTEM 2000 'INSERT' command which instructs SYSTEM 2000 to take its input from the WORKING-STORAGE of the program rather than from an input file. Construction of the data base and the inverted pointers occurs automatically, just as with the 'LOAD' command above.

Slide 6

SECURITY becomes more significant when data is available through the flexible access tools like SYSTEM 2000. On-line direct access to company files suggests the need for tight administrative controls. SYSTEM 2000 provides the security to authorize access at the field level. Each field can be assigned specific authority to be retrieved, updated, used as a qualifier for retrieval, or used as a qualifier for updating for each valid password. This feature of SYSTEM 2000 allows a data entry clerk to be authorized to retrieve and update only those fields which are pertinent to her area of responsibility. Any unauthorized access will be rejected with appropriate messages from SYSTEM 2000.

Slide 7

Back-up and Recovery capabilities are available with SYSTEM 2000. An optional logging file is used to record the results of all update activities. Operations which damage the data base will always be recognized and appropriate warnings given to the user.

Slide 8

Recovery of a damaged data base is a simple process. SYSTEM 2000 begins with a back-up copy of the data base and applies the contents of the logging file until the data base is current up to the point

Slide 8 (Cont'd)

before the error occurred. The user may optionally specify to recover only partially from the log file or to selectively apply the contents of the log file to the data base.

Slide 9

The primary purpose of a data management system is to provide better access to the data than conventional methods allow. Better accesses are measured by efficiency of access time, flexibility of getting to the data in a variety of ways, less development time for new applications, direct access to the data by non-DP-oriented personnel, and the level of technical expertise which the system demands. Of major consideration are the tradeoffs between organizing storage for efficient direct access to any combination of data and yet retaining the capability for low-cost volume updating.

Slide 10

The Immediate Access module of SYSTEM 2000 is designed to provide flexible direct access to the data using user-oriented English language instructions. The language is designed to allow the user to request any combination of data fields, sorted by any combination of values, and qualified by any logical set of conditions. Knowledge of storage structure, pointer relationships, and computer hardware are minimized.

Slide 11

The syntax of a SYSTEM 2000 command follows much the same logical pattern of English. SYSTEM 2000 is an interpretive processor; the command will be scanned, interpreted, and performed with the requested output returned to the user.

Slide 12

This slide shows a summary of the SYSTEM 2000 Immediate Access language. Both update and retrieval commands are provided. System functions are included which allow retrieval of summarized data rather than detail. The user may request sums, averages, standard deviations, maximums, minimums, or counts of data occurrences. The user may also use standard arithmetic notation (+,-,*,/) including

Slide 12 (Cont'd)

parentheses for combining detail fields in any arithmetic expression. Since SYSTEM 2000 allows any field to be designated as a key field, the user may specify a combination of any key fields as conditions for retrieval or update. These multiple conditions can be combined with boolean operators AND, OR, NOT and parenthesis to form as complex a selection creteria as the user desires.

Slide 13

Any combination of SYSTEM 2000 commands can be stored as a macro called a "STRING". Once the macro has been defined, it becomes a part of the data base itself until removed. Repeated users of the macro need only supply the macro name and the dynamic parameters which are to be used in executing the macro.

Slide 14

Arithmetic functions can also be stored for repeated use. The user may request a function by name and execute a complex arithmetic equation. Output reports can contain data which is not stored in the data base but is calculated from the contents of other fields at the time the report is produced.

Slide 15

The need for low-cost volume updating normally requires a compromise in reporting flexibility. The user usually must choose one and lose the other or pay dearly for the other. SYSTEM 2000 provides two separate features which the user may selectively choose depending on the task. The Immediate Access feature allows flexible efficient to any portion of the data. The Queue Access feature allows efficient volume updating. Both operating modes are available from an interactive terminal or within a user-written program.

Slide 16

The syntax of the Queue Access feature includes capabilities for sequential searching. Qualification of data using the inverted files requires the field to be a 'key' field and accesses to the data are direct. The 'IF' clause allows the user to search on fields which are not necessarily key. If the user wished to retrieve

Slide 16 (Cont'd)

everyone in the Data Processing Department who had worked more than ten years but only the department field was key, SYSTEM 2000 would select those employees in the Data Processing Department using the inverted table values for department and sequentially pass the selected employees checking to see if the non-key work-date was less than ten years ago.

A new capability of threshold logic is also available. The user may specify a list of conditions and require that any one of those conditions be met. These conditions lists can also be nested in parenthesis with any one of the lists to be satisfied.

Slide 17

Realizing that the user may already have a major committment to a large library of application programs or that he has special needs not available within the languages of SYSTEM 2000, an interface with procedural languages is available. This allows any COBOL program to interface with a SYSTEM 2000 data base for retrieval or updating.

Slide 18

The language available within a user program is similar to that already illustrated in the native SYSTEM 2000 languages. Each of these commands will begin in column seven of the COBOL code. A precompiler is provided which interprets these commands and substitutes standard subroutine calls before calling the COBOL compiler. This method eliminates the need for the user to establish communication constants, operation identifiers, or special links between his program instruction and SYSTEM 2000. The language provides flexible access including boolean logic for combining qualifiers.

Slides 19 and 20

It is hoped that the availability of an access tool as powerful as SYSTEM 2000 will introduce new ways of management use of data processing files. Regularly scheduled volume reporting should eventually be replaced with direct access to up-to-the-minute current data on an as-needed basis. Instead of requiring the entire file to be available on paper so that any piece of data can be found, the user should begin to depend upon the availability of current selected data at the time it is needed.

Slides 19 and 20 (Cont'd)

Two broad capabilities are available within SYSTEM 2000 for producing finished reports appropriately formatted for submitting to management.

1. The Immediate Access Module of SYSTEM 2000 includes the 'LIST' command. The user is allowed to specify page headings, column widths, number of lines per page, column headings. Page numbering and current date are generated by SYSTEM 2000. Reports can be generated with no programming and no delays.

2. A complete Report Writer comes as an additional feature of SYSTEM 2000 and provides a full range of reporting capabilities. Column and rom totals, selective exclusion of data values, replacement of codes with descriptive data, sorting, multiple reports with a singlepass of the data, and broad capabilities for segmenting a report are included in this feature.

III. Universities and Colleges Using SYSTEM 2000

A. University of Minnesota (CDC/SCOPE and KRONOS 6400 and 6600)

At the University of Minnesota (CDC/SCOPE and KRONOS 6400 and 6600) of a DBMS course, which is part of the MBA-MIS degree program. Students see the advantages of tree structures over fixed length records, and DBMS facilities in general.

SYSTEM 2000 is also used as the nucleus for a land management system which supplies the Minnesota legislature with information on the status of various land parcels.

B. University of Indiana (IBM 370/155)

SYSTEM 2000 was recently selected by Indiana following an exhaustive benchmark of the leading data base management systems. SYSTEM 2000 will be used by the Indiana University Administration for business applications, as part of its planned 5-6 year MIS effort. A data dictionary will be built first, and then a group of interelated data bases will be created, containing data on

(1) payroll and accounting
(2) student records
(3) courses
(4) inventories
(5) employees
(6) faculty/staff
(7) alumni

The interelation of the data bases will be a function of both the tree structure and networking capabilities of SYSTEM 2000.

C. University of Texas (CDC/SCOPE 6600)

At the University of Texas, SYSTEM 2000 is used by both the students and faculty for a variety of data base applications.

Among them are the following:

1. ERIC (See attached description)
2. Microbiology Department
   This data base is used to find likely candidates for rat breeding for geneology research. Data is recorded on rat genes, blood type, history, etc., and researchers use the data base to locate rats desired for a specific experiment.
3. History of Education Department
   This department uses a data base to reference information on legislation passed by the Texas Legislative during the period 1919-1967.
4. Geology Department
   Data from core samples at various ocean drilling sites and strip mines is stored in this data base.
5. Architecture Department
   This department uses a data base for area planning.
6. Anthropology Department
   Data used for study of social and economic patterns in a Mexican-Indian village is maintained in this data base.
7. Business School
   The business school has two SYSTEM 2000 data bases. The first is a portfolio management data base which is oriented toward comparing the relative success of student teams managing the portfolios. The second data base represents a hypothetical company which is used by information science and accounting students.

D. Other Universities

SYSTEM 2000 is installed at Central Texas College (IBM/VS2/TP), and will be installed at NYU (CDC/6400 - Scope) soon.

ERIC

at

The University of Texas at Austin

Through a cooperative effort of the Library and Computation Center of The University of Texas at Austin and MRI Systems Corporation of Austin, Texas, an on-line bibliographic search facility has been established for the use of students, faculty and staff. Data bases currently available are ERIC (Education), SPIN (Physics and Astronomy) and Psychological Abstracts.

The data bases were established on the Computation Center's CDC 6600/6400 computer system using SYSTEM 2000®, a proprietary data management system of MRI*. Using the "string" facility of SYSTEM 2000, a mnemonic language was developed to permit search and display of bibliographic information on documents in the retrieval system. No knowledge of the SYSTEM 2000 language is required or assumed. Search keys were chosen to be the descriptors assigned by the data base supplier, year of publication, and author's surnames.

The query language is described on the following pages. The user is forced by the syntax of the language to determine the number (count) of articles retrieved before displaying the bibliographic information on a low speed terminal or on a high speed printer. This feature helps the user to narrow his request to a "reasonable" number before printing.

All data bases mentioned above use the same query language. This enables users to switch from one data base to another with ease.

---

*For further information, contact MRI Systems Corporation, P.O. 9968, Austin, Texas 78766 (512) 258-5171

INSTRUCTIONS FOR USING THE COMPUTERIZED ERIC DATA BASE

INTRODUCTION:

The ERIC collection of educational materials contains two types of materials. The RIE (Reports in Education) materials are for the most part unpublished reports whose complete text can be found in the microfiche collection in the Education and Psychology Library. The CIJE (Current Index to Journals in Education) materials have been published and the complete text can be found in the original journals.

The computerized ERIC data base should be looked upon as an automated index to recent ERIC accessions. The data base contains references to all RIE documents numbered ED040431 and following and to all CIJE articles numbered EJ018072 and following.

Each item in the ERIC collection has been indexed by the ERIC clearinghouses by assigning to it a number of "descriptors". The descriptors indicate the subject content of the item. The printed index to the ERIC collection contains references to only about half of the descriptors assigned to an item. Since the computerized ERIC data base contains references to all the descriptors for an item, a much more complete index is available using the automated index.

The valid descriptors are listed in the "Thesaurus of ERIC Descriptors". Only the descriptors listed in the thesaurus may be used as retrieval keys -- NO others are valid.

CONTENT:

The ERIC data base is intended to be an automated index to the RIE microfiche collection and to the journals covered by CIJE. For this reason and to provide the user with as broad a coverage as possible, only the first sixty characters of each title are included and the abstracts have not been included.

Each component (title, descriptor, etc.) of an item is identified by an identifying code number followed by a $. The code number for each item as a whole is 0. The other components and their identifying code numbers are:

1$	ERIC accession number (microfiche number for RIE items)
11$	Journal name (if published)
13$	Journal volume, issue number, page
17$	Year of publication or issue
21$	Title (if the title has been truncated at sixty characters, an * appears as the last character)
31$	Author's surname
33$	Author's forenames
41$	ERIC descriptors assigned (from the ERIC thesaurus)
51$	Identifiers assigned to the item (free text)

ACCESS:

The basic method of access to an item is via the ERIC descriptors assigned to that item. Since the results of your search will be printed on a relatively slow typewriter-type device, you must determine the amount of material to be printed. For example, a search question might be too broad and retrieve thousands or hundreds of items. A practical limit to the number of items to be displayed on the terminal is about twenty. To obtain the number of items which were retrieved in response to your search request, a series of "count" (abbreviated CNT) commands are available.

Suppose you wish to retrieve items which have been assigned a particular ERIC descriptor , say MAN MACHINE SYSTEMS. First, a "count" command is typed which specifies this one descriptor

```
$CNT1  (MAN MACHINE SYSTEMS) Ⓡ
```

Note that each command is preceded by a $, the descriptor is enclosed in parentheses, and the "carriage return" button (indicated by Ⓡ) is pushed after <u>each</u> command. This is a rather broad search since every item which was assigned this descriptor will be retrieved. The response to your command might be

```
CNT O$  246
```

indicating that 246 items were retrieved. This is probably more items than are desired and the search should be narrowed by requesting the "count" of items which have been assigned two descriptors of interest, say, INTERACTIONS, as well as MAN MACHINE SYSTEMS. In other words, we want the "count" of items which have been assigned <u>both</u> the descriptors INTERACTIONS <u>and</u> MAN MACHINE SYSTEMS. The appropriate command is

```
$CNT2  (INTERACTION, MAN MACHINE SYSTEMS) Ⓡ
```

Note again that the command is preceded by a $, the descriptors are enclosed in parentheses and separated by a comma, and a "carriage return" follows the command. The order of descriptors is irrelevant. The response might be

```
CNT O$   2
```

indicating that the more restrictive query (requiring the presence of two specific descriptors) has narrowed the search to two items.

At this point, specific information on each of the retrieved items may be requested. Several commands are available for this purpose. There are five commands for displaying the information at the terminal, $SHOW-A$, $SHOW-B$, ..., $SHOW-E$, each "show" command giving a different set of components. For example, the following command (issued after a count command has retrieved the desired number of items)

```
$SHOW-A$ Ⓡ
```

will display the accession number and title of each item, e.g.,

```
1$  ED060891
21$  INTERACTIVE BIBLIOGRAPHIC SEARCH: THE USER/COMPUTER INTERFACE:*
1$  ED061052
21$  MAN: PLANETARY DISEASE.  THE 1971 B.Y. MORRISON MEMORIAL LECTU*
```

while the command

```
$SHOW-D$ Ⓡ
```

would print the accession number and title as well as the year, publication information, authors, and all descriptors, e. g.,

```
1$   ED060891
21$   INTERACTIVE BIBLIOGRAPHIC SEARCH: THE USER/COMPUTER INTERFACE;*
17$  71
  31$  WALKER
  33$  DONALD E., ED.
  41$  BIBLIOGRAPHIES
  41$  CONFERENCE REPORTS
  41$  INFORMATION RETRIEVAL
  41$  INTERACTION
  41$  MAN MACHINE SYSTEMS
  41$  ON LINE SYSTEMS
  41$  SEARCH STRATEGIES
  41$  WORKSHOPS
1$   ED061052
21$  MAN:  PLANETARY DISEASE.  THE 1971 B.Y.MORRISON MEMORIAL LECTU*
17$  71
  31$  MCHARG
  33$  IAN L.
  41$  ECOLOGY
  41$  ENVIRONMENTAL INFLUENCES
  41$  EVOLUTION
  41$  HUMAN ENGINEERING
  41$  INTERACTION
  41$  MAN MACHINE SYSTEMS
  41$  SPEECHES
```

The general access procedure is to issue "count" commands with the list of descriptors in parentheses and then an appropriate "show" command when the number of items retrieved by the "count" command is reasonable.

If it is desired to have your items printed by the line printers at the Computation Center rather than at the terminal (useful for long lists), the command $PRINT followed by your name enclosed in parentheses will accomplish this. For example, the command might be

```
$PRINT (JOHN Q. PUBLIC)  Ⓡ
```

AVAILABLE COMMANDS:

There are three "count" commands for one, two, and three descriptors respectively. The syntax of these commmands is

```
$CNT1  (descriptor-1)  Ⓡ
$CNT2  (descriptor-1, descriptor-2)  Ⓡ
$CNT3  (descriptor-1, descriptor-2, descriptor-3)  Ⓡ
```

AVAILABLE COMMANDS (continued)

These three commands retrieve and count the items which have been assigned all the descriptors listed between parentheses.

There are five "show" commands. Their syntax and purpose is

```
$SHOW-A$  Ⓡ  ...Prints accession number and title.
$SHOW-B$  Ⓡ  ...Same as above plus publication information and year.
$SHOW-C$  Ⓡ  ...Same as above plus author information.
$SHOW-D$  Ⓡ  ...Same as above plus all descriptors.
$SHOW-E$  Ⓡ  ...Same as above plus all identifiers.
```

There are several commands which enable the user to add a condition to the previously executed "count" command. The command

```
$ADD-AUT (author-surname)  Ⓡ
```

retrieves the subset of the items just previously retrieved which were written by the author specified in parentheses. There are three commands which enable the user to specify the year or years of publication which are of interest. These commmands retrieve a subset of the items just previously retrieved. Their syntax and purpose is

```
$ADD-YEAR-EQUAL (year)  Ⓡ  ...Narrows the search to the specified year.
$ADD-YEAR-EARLIER (year)  Ⓡ  ...Narrows the search to the specified
                                 year and earlier.
$ADD-YEAR-LATER (year)  Ⓡ  ...Narrows the search to the specified
                                 year and later.
```

NOTE: Only the last two digits of the year should be typed, e.g.,69.

It is also possible to narrow the search by adding one more descriptor by issuing the command

```
$ADD-TERM (descriptor)  Ⓡ
```

This method is faster than issuing another "count" command.

As mentioned before, the command

```
$PRINT (user's-name)  Ⓡ
```

will cause the retrieved items to be printed at the Computation Center's printers. This command should be used if the "count" of desired items is large.

SYSTEM 2000 COMMANDS:

The ERIC data base is implemented using a general purpose data management program known as SYSTEM 2000. Besides the special commands discussed above, the entire SYSTEM 2000 retrieval language is available. See the SYSTEM 2000 Reference Manual for details.